

МОНОГРАФИЯ БАЗОВЫЙ ПРОЕКТ KLT-RBD

Том VIII. KLT-RBD5.10: программы, базы данных и машинная архитектура

Расширенная редакция v180: software/database layer, TextWork, import/export, proof-ledger dashboard, API, rollback, site-safe release

Автор идей, конструкций, терминов, программной линии и реперной базы данных: Курпишев Иван Борисович / Ivan Borisovich Kurpishev, Independent Researcher, Kaliningrad.

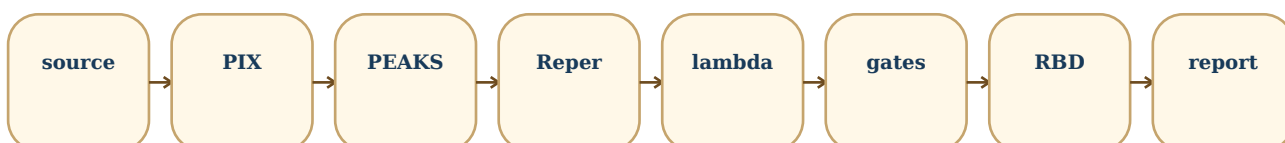
```
status = volume8_software_database_expanded_without_truth_status
file = mbp180.pdf
short paths only: mbp180/d/mbp180.pdf, mbp180/src/mbp180.md
```

No-loss rule for this volume: no earlier monograph, article, package, registry, site page, proof ledger, blocker card or RBD layer is cancelled by this draft. The present volume is an expanded working redaction. It carries previous material forward and marks what must later be inserted in full text form.

Layer	Role in the expanded monograph
Master corpus 5.0	C@C, Reper, lambda, delta_truth, NAPG 3.0, KLT/RBD
Monograph 5.1	no-loss layer, RBD/RPD, KLT 4.14, KLT 5.1, predictive Reper manifolds
PILOT-01	formula-chain audit, gap-node, Fano barrier, theorem-candidate discipline
SIGMA/NAPG	stratified time, Pack, Hodge, associator amplitude, reduced obstruction quotient
v32-v179	physics, chemistry, biology, DNA, software, site and artifact development ledger
External official routes	NIST ASD, PubChem PUG-REST, NCBI Datasets, Ensembl REST, UniProt, RCSB PDB, arXiv API, EIS procurement

Том VIII строится не как сокращённая справка, а как рабочий слой полной компиляции. В дальнейшем этот текст должен быть соединён с исходными PDF, HTML, TeX, DOCX, SQLite, ZIP и site-архивами через единый source atlas. Там, где прежний материал уже существует, новая редакция не переписывает его насильственно, а даёт пояснение, уточнение статуса и место вставки в полную книжную сборку.

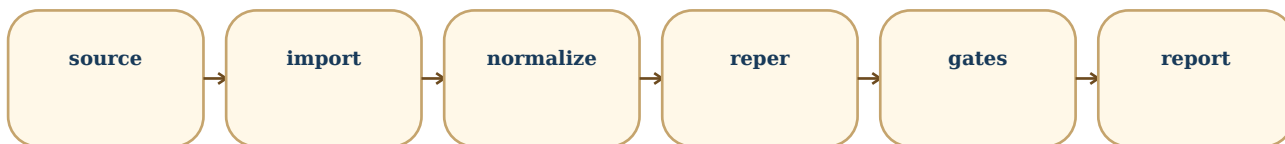
Ключевой запрет: удобная формула, красивая аналогия, похожая структура, программный отчёт или удачный график не получают повышенный статус без D, Dom, proof-object, Evidence-D, matched-null, blind gate, Fano/morphism gate and reproduction gate. Иначе монография снова превратится в коллекцию впечатляющих заявлений, а мы вроде бы собираем математику, не витрину для восторгов.



Оглавление рабочей редакции

1. Архитектура KLT-RBD5.10 как программного слоя монографии
2. SQLite-first ядро и активная реперная база
3. Input layer: документы, таблицы, сметы, статьи, архивы
4. TextWork parser и claim-status extraction
5. PIX и PEAKS в программной реализации
6. Reper builder: R,I,U,D в машинной карточке
7. Lambda scanner и delta_truth в отчёте
8. CGI* как blocker-veto index
9. Proof-ledger и dependency graph
10. Evidence-D gates в программном конвейере
11. Dashboard как интерфейс blockers, а не витрина побед
12. Import/export command set
13. API и интеграция с внешними платформами
14. Security, hash, rollback and audit trail
15. Роль APK/Flutter SDK
16. Тестовые сценарии и smoke-test corpus
17. Формат отчёта для не-математика
18. Roadmap промышленной версии

1. Архитектура KLT-RBD5.10 как программного слоя монографии



Архитектура KLT-RBD5.10 как программного слоя монографии рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри software/database каждый объект должен быть описан как событие@состояние, получить $\text{Reper}(R,I,U;D)$, быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - показать программу как продолжение математической базы, а не отдельное приложение. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Архитектура KLT-RBD5.10 как программного слоя монографии" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Rep}(R,I,U;D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене software/database эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Архитектура KLT-RBD5.10 как программного слоя монографии" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась

исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для software/database пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Архитектура KLT-RBD5.10 как программного слоя монографии" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смешение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Архитектура KLT-RBD5.10 как программного слоя монографии" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Архитектура KLT-RBD5.10 как программного слоя монографии": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже

задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

2. SQLite-first ядро и активная реперная база

SQLite-first ядро и активная реперная база рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри database каждый объект должен быть описан как событие@состояние, получить $\text{Reper}(R, I, U; D)$, быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - объяснить, почему база не является простой библиографией. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "SQLite-first ядро и активная реперная база" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Reper}(R, I, U; D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене database эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "SQLite-first ядро и активная реперная база" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для database пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем

показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть `review_seed`, `hypothesis`, `theorem_candidate` или `blocker`. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как `source atlas`. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, `proof route`, исторический комментарий или машинный `artefact`. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "SQLite-first ядро и активная реперная база" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: `source -> extraction -> Reper -> gates -> graph -> report`. Схема не является украшением; она показывает, где именно возможны ошибки: потеря `source`, смещение `Dom`, отсутствие `D`, неверный `null`, ложный `morphism`, невалидный `proof-object` или неподтверждённый `reproduction gate`.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, `rollback`-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "SQLite-first ядро и активная реперная база" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие `proof-blockers` закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как гряда папок после торнадо.

Итоговая фиксация раздела "SQLite-first ядро и активная реперная база": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а `prior art` остаётся `prior art`.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

Таблица	Назначение
<code>source_register</code>	источник и его контрольные хэши

Таблица	Назначение
observation	нормализованные наблюдения
reper_card	R,I,U,D и status
evidence_gate	D/Dom/null/blind/proof/repro
status_ledger	история решений

3. Input layer: документы, таблицы, сметы, статьи, архивы

Input layer: документы, таблицы, сметы, статьи, архивы рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри import каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - описать входные форматы и правила нормализации. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Input layer: документы, таблицы, сметы, статьи, архивы" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись C@C -> Rep(R,I,U;D). R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене import эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Input layer: документы, таблицы, сметы, статьи, архивы" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен

догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для import пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Input layer: документы, таблицы, сметы, статьи, архивы" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Input layer: документы, таблицы, сметы, статьи, архивы" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Input layer: документы, таблицы, сметы, статьи, архивы": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие

исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

4. TextWork parser и claim-status extraction

TextWork parser и claim-status extraction рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри textwork каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - показать переход от текста к карточкам утверждений. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "TextWork parser и claim-status extraction" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись C@C -> Rep(R,I,U;D). R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене textwork эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "TextWork parser и claim-status extraction" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для textwork пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая

педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть `review_seed`, `hypothesis`, `theorem_candidate` или `blocker`. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как `source atlas`. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, `proof route`, исторический комментарий или машинный `artefact`. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "TextWork parser и claim-status extraction" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: `source -> extraction -> Reper -> gates -> graph -> report`. Схема не является украшением; она показывает, где именно возможны ошибки: потеря `source`, смешение `Dom`, отсутствие `D`, неверный `null`, ложный `morphism`, невалидный `proof-object` или неподтверждённый `reproduction gate`.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, `rollback`-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "TextWork parser и claim-status extraction" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие `proof-blockers` закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как грудa папок после торнадо.

Итоговая фиксация раздела "TextWork parser и claim-status extraction": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а `prior art` остаётся `prior art`.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

Редакционная помета полной версии: в финальной компиляции сюда добавляются прямые вставки из прежних монографий, статей, таблиц RBD и карточек доказательного реестра. Текущая версия не удаляет эти материалы, а задаёт место и статус их интеграции.

5. PIX и PEAKS в программной реализации

PIX и PEAKS в программной реализации рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри software ontology каждый объект должен быть описан как событие@состояние, получить $\text{Reper}(R,I,U;D)$, быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - связать первичные элементы и устойчивые состояния. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "PIX и PEAKS в программной реализации" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Reper}(R,I,U;D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене software ontology эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "PIX и PEAKS в программной реализации" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для software ontology пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но

понижает её статус. Она может быть `review_seed`, `hypothesis`, `theorem_candidate` или `blocker`. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как `source atlas`. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, `proof route`, исторический комментарий или машинный `artefact`. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "PIX и PEAKS в программной реализации" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: `source -> extraction -> Reper -> gates -> graph -> report`. Схема не является украшением; она показывает, где именно возможны ошибки: потеря `source`, смещение `Dom`, отсутствие `D`, неверный `null`, ложный `morphism`, невалидный `proof-object` или неподтверждённый `reproduction gate`.

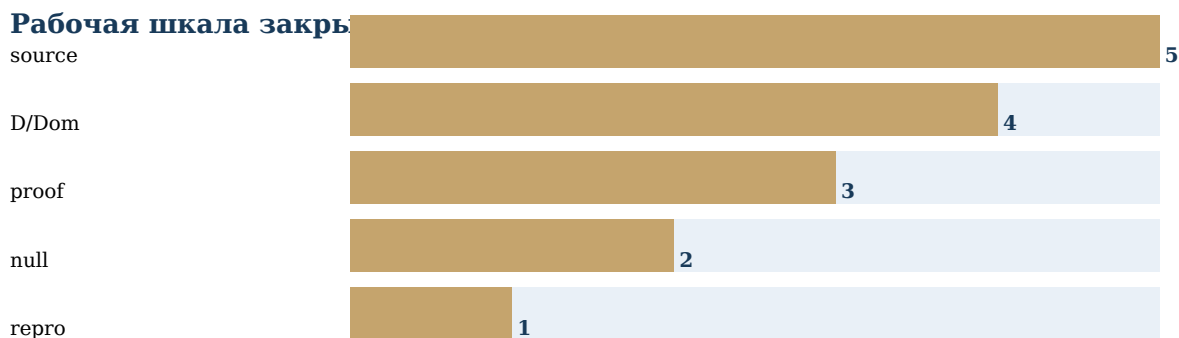
Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, `rollback`-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "PIX и PEAKS в программной реализации" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие `proof-blockers` закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как гряда папок после торнадо.

Итоговая фиксация раздела "PIX и PEAKS в программной реализации": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а `prior art` остаётся `prior art`.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.



6. Reper builder: R,I,U,D в машинной карточке

Reper builder: R,I,U,D в машинной карточке рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри machine reper каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - описать поля карточки и ошибки заполнения. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Reper builder: R,I,U,D в машинной карточке" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись C@C -> Rep(R,I,U;D). R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене machine reper эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Reper builder: R,I,U,D в машинной карточке" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для machine reper пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Reper builder: R,I,U,D в машинной карточке" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смешение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Reper builder: R,I,U,D в машинной карточке" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Reper builder: R,I,U,D в машинной карточке": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся

prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

```
ReperCard = (source_id, unit_id, R, I, U, D, Dom, lambda, delta_truth, blockers, status)
```

7. Lambda scanner и delta_truth в отчёте



Lambda scanner и delta_truth в отчёте рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри lambda engine каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - развести структурную близость и доказательный статус. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Lambda scanner и delta_truth в отчёте" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись C@C -> Rep(R,I,U;D). R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене lambda engine эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше

не продолжать.

Четвёртый уровень - доказательный. Для раздела "Lambda scanner и delta_truth в отчёте" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для lambda engine пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Lambda scanner и delta_truth в отчёте" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Lambda scanner и delta_truth в отчёте" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Lambda scanner и delta_truth в отчёте": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

8. CGI* как blocker-veto index

CGI* как blocker-veto index рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри status engine каждый объект должен быть описан как событие@состояние, получить $\text{Reper}(R,I,U;D)$, быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - не дать индексу стать заменой доказательства. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "CGI* как blocker-veto index" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Rep}(R,I,U;D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене status engine эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "CGI* как blocker-veto index" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для status engine пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "CGI* как blocker-veto index" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "CGI* как blocker-veto index" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "CGI* как blocker-veto index": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

Редакционная помета полной версии: в финальной компиляции сюда добавляются прямые вставки из прежних монографий, статей, таблиц RBD и карточек доказательного реестра. Текущая версия не удаляет эти материалы, а задаёт место и статус их интеграции.

9. Proof-ledger и dependency graph

Proof-ledger и dependency graph рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри proof system каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - показать, как программа хранит доказательные зависимости. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Proof-ledger и dependency graph" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись C@C -> Rep(R,I,U;D). R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене proof system эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Proof-ledger и dependency graph" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для proof system пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем

показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть `review_seed`, `hypothesis`, `theorem_candidate` или `blocker`. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как `source atlas`. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, `proof route`, исторический комментарий или машинный `artefact`. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Proof-ledger и dependency graph" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: `source -> extraction -> Reper -> gates -> graph -> report`. Схема не является украшением; она показывает, где именно возможны ошибки: потеря `source`, смешение `Dom`, отсутствие `D`, неверный `null`, ложный `morphism`, невалидный `proof-object` или неподтверждённый `reproduction gate`.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, `rollback`-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Proof-ledger и dependency graph" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие `proof-blockers` закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как гряда папок после торнадо.

Итоговая фиксация раздела "Proof-ledger и dependency graph": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а `prior art` остаётся `prior art`.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

10. Evidence-D gates в программном конвейере

Evidence-D gates в программном конвейере рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри evidence каждый объект должен быть описан как событие@состояние, получить $\text{Reper}(R, I, U; D)$, быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - связать D/Dom, null, blind, proof и reproduction. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Evidence-D gates в программном конвейере" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Reper}(R, I, U; D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене evidence эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Evidence-D gates в программном конвейере" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для evidence пример строится так: берётся source-object, выделяются PИХ-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё

недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Evidence-D gates в программном конвейере" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

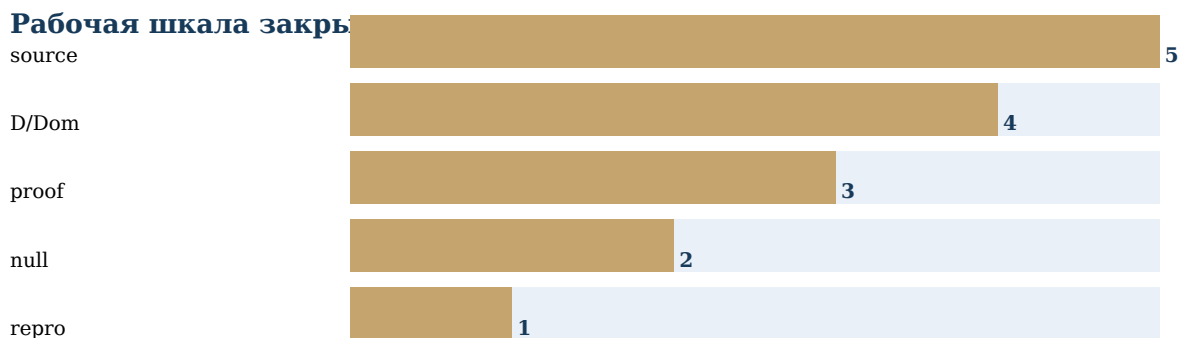
Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Evidence-D gates в программном конвейере" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Evidence-D gates в программном конвейере": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.



11. Dashboard как интерфейс blockers, а не витрина побед

Dashboard как интерфейс blockers, а не витрина побед рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри dashboard каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - объяснить честную отчётность для пользователей. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает gap или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Dashboard как интерфейс blockers, а не витрина побед" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Reper}(R,I,U;D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене dashboard эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше

не продолжать.

Четвёртый уровень - доказательный. Для раздела "Dashboard как интерфейс blockers, а не витрина побед" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для dashboard пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Dashboard как интерфейс blockers, а не витрина побед" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Dashboard как интерфейс blockers, а не витрина побед" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

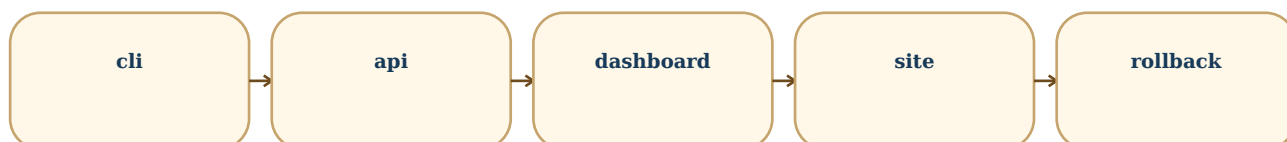
Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Dashboard как интерфейс blockers, а не витрина побед": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается.

Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

12. Import/export command set



Import/export command set рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри cli каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - описать команды KLT-RBD5.10 без имитации готовой промышленной системы. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает gap или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Import/export command set" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись C@C -> Rep(R,I,U;D). R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене cli эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше

не продолжать.

Четвёртый уровень - доказательный. Для раздела "Import/export command set" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для cli пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Import/export command set" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Import/export command set" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Import/export command set": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

```
klt-rbd510 init-db
klt-rbd510 import --source package.zip
klt-rbd510 normalize
klt-rbd510 build-reper
klt-rbd510 scan-lambda
klt-rbd510 proof-ledger
klt-rbd510 export-report
```

Редакционная помета полной версии: в финальной компиляции сюда добавляются прямые вставки из прежних монографий, статей, таблиц RBD и карточек доказательного реестра. Текущая версия не удаляет эти материалы, а задаёт место и статус их интеграции.

13. API и интеграция с внешними платформами

API и интеграция с внешними платформами рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри `api` каждый объект должен быть описан как событие@состояние, получить `Reper(R,I,U;D)`, быть помещён в `Dom` и пройти `proof/status-gates`. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - подготовить мост к ЕИС, 1С, корпоративным аудитам и научным базам. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется `status ladder`, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится `source-layer` или `appendix-layer`.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает `gap` или `blocker`; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического `prior art`, а в соединении классических структур, источниковых документов, `Reper`-карт, `Evidence-D` и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "API и интеграция с внешними платформами" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет `Dom`, мы не знаем, где допустим вывод. Если объект не имеет `D`, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись `C@C -> Rep(R,I,U;D)`. `R` удерживает установленное содержание, `I` - инвариант или проектную идею, `U` - поле допустимых вариантов, `D` - достаточное основание. В домене `api` эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на `raw layer`, карту преобразования и контрольный `hash`. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет

говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "API и интеграция с внешними платформами" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для api пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "API и интеграция с внешними платформами" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "API и интеграция с внешними платформами" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "API и интеграция с внешними платформами": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся

prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

14. Security, hash, rollback and audit trail

Security, hash, rollback and audit trail рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри security каждый объект должен быть описан как событие@состояние, получить $\text{Reper}(R,I,U;D)$, быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - сохранить воспроизводимость и защиту от фальсификаций. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Security, hash, rollback and audit trail" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Rep}(R,I,U;D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене security эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Security, hash, rollback and audit trail" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для security пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Security, hash, rollback and audit trail" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Security, hash, rollback and audit trail" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Security, hash, rollback and audit trail": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

15. Роль APK/Flutter SDK

Роль APK/Flutter SDK рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри mobile каждый объект должен быть описан как событие@состояние, получить $\text{Reper}(R,I,U;D)$, быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - показать мобильную и демонстрационную ветку. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Роль APK/Flutter SDK" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Rep}(R,I,U;D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене mobile эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Роль APK/Flutter SDK" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для mobile пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но

понижает её статус. Она может быть `review_seed`, `hypothesis`, `theorem_candidate` или `blocker`. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как `source atlas`. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, `proof route`, исторический комментарий или машинный `artefact`. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Роль APK/Flutter SDK" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: `source -> extraction -> Reper -> gates -> graph -> report`. Схема не является украшением; она показывает, где именно возможны ошибки: потеря `source`, смещение `Dom`, отсутствие `D`, неверный `null`, ложный `morphism`, невалидный `proof-object` или неподтверждённый `reproduction gate`.

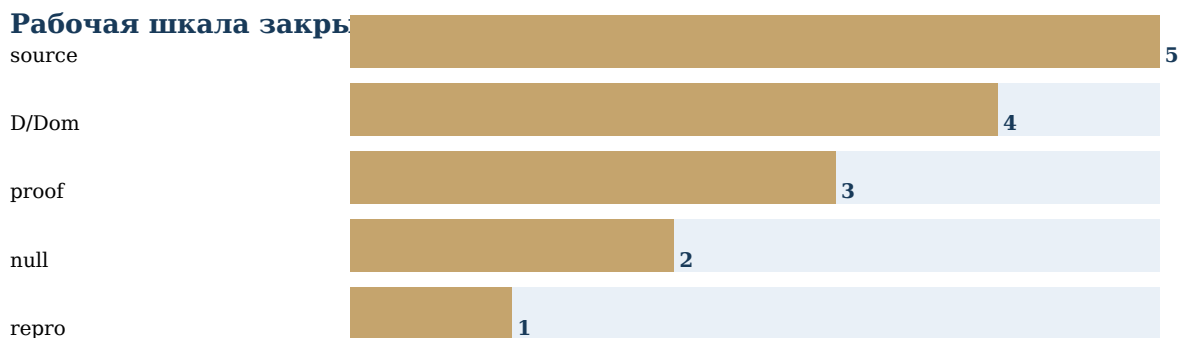
Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, `rollback`-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Роль APK/Flutter SDK" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие `proof-blockers` закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как гряда папок после торнадо.

Итоговая фиксация раздела "Роль APK/Flutter SDK": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а `prior art` остаётся `prior art`.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.



16. Тестовые сценарии и smoke-test corpus

Тестовые сценарии и smoke-test corpus рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри testing каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - не подменять тестирование красивым интерфейсом. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Тестовые сценарии и smoke-test corpus" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись C@C -> Rep(R,I,U;D). R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене testing эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Тестовые сценарии и smoke-test corpus" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для testing пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Тестовые сценарии и smoke-test corpus" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Тестовые сценарии и smoke-test corpus" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Тестовые сценарии и smoke-test corpus": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

Редакционная помета полной версии: в финальной компиляции сюда добавляются прямые вставки из прежних монографий, статей, таблиц RBD и карточек доказательного реестра. Текущая версия не удаляет эти материалы, а задаёт место и статус их интеграции.

17. Формат отчёта для не-математика

Формат отчёта для не-математика рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри report каждый объект должен быть описан как событие@состояние, получить $\text{Reper}(R,I,U;D)$, быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - сделать результаты понятными чиновнику, бухгалтеру, инженеру и рецензенту. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Формат отчёта для не-математика" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Rep}(R,I,U;D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене report эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Формат отчёта для не-математика" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers.

Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для report пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть review_seed, hypothesis, theorem_candidate или blocker. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как source atlas. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, proof route, исторический комментарий или машинный artefact. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Формат отчёта для не-математика" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: source -> extraction -> Reper -> gates -> graph -> report. Схема не является украшением; она показывает, где именно возможны ошибки: потеря source, смещение Dom, отсутствие D, неверный null, ложный morphism, невалидный proof-object или неподтверждённый reproduction gate.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, rollback-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Формат отчёта для не-математика" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие proof-blockers закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как гряда папок после торнадо.

Итоговая фиксация раздела "Формат отчёта для не-математика": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а prior art остаётся prior art.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

18. Roadmap промышленной версии

Roadmap промышленной версии рассматривается как часть полного корпуса KLT-RBD, а не как отдельная техническая заметка. Внутри roadmap каждый объект должен быть описан как событие@состояние, получить Reper(R,I,U;D), быть помещён в Dom и пройти proof/status-gates. Это неприятно медленнее, чем просто объявить результат, зато именно так появляется проверяемый корпус, а не литературная туманность.

Практическая задача раздела - определить путь от текущих пакетов к пилотам и лицензированию. Для читателя это означает движение от простого примера к формальному объекту: сначала показывается человеческий смысл, затем вводится карточка RBD, затем фиксируется status ladder, а затем указывается, какие источники и доказательства нужны для повышения статуса. Никакой материал прежних монографий при этом не исчезает: он становится source-layer или appendix-layer.

В полной редакции этот раздел должен получить три обязательных слоя: учебный пример, контрпример и доказательный протокол. Пример показывает, как KLT-RBD работает при достаточном основании; контрпример показывает, где возникает гар или blocker; протокол показывает, какие данные нужны программе KLT-RBD5.10 для машинного отчёта.

Новизна авторской линии Курпишева И.Б. состоит не в присвоении классического prior art, а в соединении классических структур, источниковых документов, Reper-карт, Evidence-D и активной базы RBD/RPD. Поэтому каждый раздел отделяет: что является известным математическим или прикладным фоном, а что является авторской архитектурой KLT-RBD.

Первый пояснительный уровень раздела "Roadmap промышленной версии" предназначен для внешнего читателя. Он должен понять не только термин, но и практическую причину его введения. Если объект не имеет источника, мы не знаем, что именно проверяем. Если объект не имеет Dom, мы не знаем, где допустим вывод. Если объект не имеет D, мы не знаем, чем этот вывод поддержан. Эта тройная защита выглядит скучно, но без неё метод быстро скатывается в красивую угадку.

Второй уровень - формальный. Здесь объект переводится в запись $C@C \rightarrow \text{Rep}(R,I,U;D)$. R удерживает установленное содержание, I - инвариант или проектную идею, U - поле допустимых вариантов, D - достаточное основание. В домене roadmap эти роли получают свои технические имена, но логика остаётся одной: никакая строка, формула, реакция, белок, смета или философская мысль не проходит дальше без паспорта основания.

Третий уровень - машинный. Программа KLT-RBD5.10 должна получить не красивый абзац, а нормализуемый объект: идентификатор источника, единицы, тип, домен, версию, ссылку на raw layer, карту преобразования и контрольный hash. Именно поэтому в полной монографии рядом с философскими объяснениями должны стоять поля базы данных. Иначе текст будет говорить одно, а программа делать неизвестно что, этот старый человеческий жанр лучше не продолжать.

Четвёртый уровень - доказательный. Для раздела "Roadmap промышленной версии" требуется proof-ledger: какие определения используются, какие зависимости уже доказаны, какие являются гипотезами, какие являются theorem-candidates, а какие являются blockers. Это особенно важно для многотомника: читатель не должен догадываться, где закончилась математика и началась исследовательская программа.

Пятый уровень - пример. В полной версии каждая идея должна иметь минимум один бытовой или инженерный пример. Для roadmap пример строится так: берётся source-object, выделяются PIX-элементы, группируются PEAKS-состояния, строится Reper-карта, затем показывается, что происходит при наличии D и что ломается при его отсутствии. Такая педагогика спасает читателя от утопления в аббревиатурах, хотя аббревиатуры, конечно, стараются.

Шестой уровень - контрпример. Если аналогичная структура обнаружена без источника, без домена, без морфизма или без внешней проверки, KLT-RBD не уничтожает идею, но понижает её статус. Она может быть `review_seed`, `hypothesis`, `theorem_candidate` или `blocker`. В этом различии и состоит зрелость проекта: не всё ценное уже доказано, но не всё недоказанное можно выдавать за результат.

Седьмой уровень - связность с предыдущими монографиями. Материал монографий 2.0, 2.1, 2.3, 5.0, 5.1, 6.0 и сборок v32-v179 должен войти в этот раздел не как хаотический архив, а как `source atlas`. Каждый прежний фрагмент получает место: формула, определение, пример, приложение, `proof route`, исторический комментарий или машинный `artefact`. Потеря фрагмента считается ошибкой сборки.

Восьмой уровень - библиография. Раздел "Roadmap промышленной версии" должен отделять классический фон от авторского вклада. Классические источники дают язык, но авторская новизна Курпишева И.Б. состоит в реперно-проективной упаковке, Evidence-D воротах, активной RBD/RPD и программной линии KLT-RBD5.10. Это разделение нужно не из скромности, а чтобы рецензенту не пришлось вытаскивать его клещами.

Девятый уровень - схема. Для полной версии здесь должна стоять диаграмма: `source -> extraction -> Reper -> gates -> graph -> report`. Схема не является украшением; она показывает, где именно возможны ошибки: потеря `source`, смещение `Dom`, отсутствие `D`, неверный `null`, ложный `morphism`, невалидный `proof-object` или неподтверждённый `reproduction gate`.

Десятый уровень - связь с сайтом. Страница проекта должна хранить не только итоговый PDF, но и карту тома, предыдущие версии, `rollback`-ссылки, исходные архивы и контрольные суммы. Сайт в проекте KLT-RBD является не витриной, а памятью корпуса. Если сайт теряет блок ФИПС, РЦИС или патенты, это не эстетическая мелочь, а повреждение доказательной биографии проекта.

Одиннадцатый уровень - практическая применимость. Раздел "Roadmap промышленной версии" должен показывать, что метод может быть проверен на реальном массиве: научном корпусе, смете, контракте, химических данных, белковых последовательностях или философской структуре аргумента. Проверка должна быть ограниченной, воспроизводимой и честной; иначе слово "предсказательный" превращается в рекламное заклинание.

Двенадцатый уровень - дорожная карта. Для следующей редакции нужно указать, какие источники добавляются полностью, какие таблицы RBD присоединяются, какие рисунки рисуются заново, какие `proof-blockers` закрываются и какие внешние маршруты проверяются. Полная монография должна расти как дерево с корнями, а не как груда папок после торнадо.

Итоговая фиксация раздела "Roadmap промышленной версии": материал сохраняется, уточняется, расширяется и получает место в общей архитектуре. Ничего из прежнего корпуса не вычёркивается молча. Непроверенное не повышается. Авторство Курпишева Ивана Борисовича указывается для авторских конструкций, а `prior art` остаётся `prior art`.

Редакционная задача дальнейшей сборки: заменить краткие служебные вставки прямыми расширенными фрагментами из всех прежних монографий, статей и пакетов. Эта версия уже задаёт ткань будущей полной книги, но следующая стадия должна переносить большие исходные главы без потерь, с нормальной пагинацией, оглавлением, иллюстрациями, словарём и индексацией.

Словарь тома

Термин	Пояснение
KLT-RBD5.10	Программная линия импорта, нормализации, Reper-карт, доказательных ворот, отчёта и сайта.
TextWork	Слой машинного разбора текста на утверждения, определения, blockers and proof routes.
Dashboard	Интерфейс, который показывает статусы и blockers, но не повышает их сам.
Rollback	Воспроизводимое возвращение к сохранённой точке сборки.
Hash ledger	Контрольная цепочка файлов, источников и отчётов.

Библиографическая и внешняя карта

Источник	Роль в томе
Monograph 5.0	Master corpus: C@C, Reper, lambda, NAPG 3.0, KLT/RBD.
Monograph 5.1	No-loss layer, RBD/RPD, KLT 4.14/5.1 and predictive Reper manifolds.
PILOT-01	Formula-chain audit, Fano barrier, gap discipline.
SIGMA/NAPG	Stratified time, Pack, Hodge, associator rigidity.
NIST ASD	Official atomic spectra route for physical Evidence-D.
PubChem PUG-REST	URL-based official programmatic chemistry route.
NCBI / Ensembl / UniProt / RCSB	Biology and DNA source-bound routes.
arXiv API	External full-text / metadata corpus route for independent tests.
ЕИС Госзакупки	Official procurement source route for applied audit.

Статусная фиксация: truth_status_promoted_count = 0; publication_verified_status_count = 0. Все повышающие утверждения требуют отдельного доказательного протокола и внешней проверки.