

Что выполняет ВЕРДИКТ и что остаётся за вашей системой

Модель исполнения, граница ответственности, версии, проверка до выпуска, сбой и отмена, внешний решатель, развёртывание и доступ, гарантии. Восемь модулей, каждый читается отдельно.

- D1 Модель исполнения: сессия, такт, контракт данных, таблицы решений, цели, стратегия выбора, поручения, сроки, обратные действия
- D2 Интеграционная граница: кто запускает, кто вызывает, где хранится состояние, кто отсчитывает сроки
- D3 Версии и история правок: что воспроизводится и как сравниваются редакции
- D4 Проверка до выпуска: случаи, противоречия, пробелы, находки на стыках, охват
- D5 Сбой и обратные действия: исход отката, пределы, повторная доставка
- D6 Внешний решатель: как вычисляется граница полномочий до обращения к модели
- D7 Развёртывание, безопасность и доступ: что остаётся в вашем контуре
- D8 Гарантии: на что можно положиться

Георгий Бах

george.bv@verdikt-app.ru · verdikt-app.ru

D1 Модель исполнения

сессия	Одна операция процесса: своё состояние, своя запись и своя версия правил — та, на которой операция началась.
такт	Один цикл исполнения: процесс возобновляется, вычисляет правила по текущим данным и выбирает следующий шаг из допустимых, а не идёт по заранее нарисованному маршруту.
контракт данных	Всё, с чем работает процесс, объявлено типами: состояние, факты, поручения наружу. Чего нет в контракте, того для процесса не существует.
таблицы решений	Решение раскладывается на читаемые ступени, таблицы ссылаются друг на друга. Условие — фраза, а не код, и его разбирают формально. Логика шагов пишется рядом, под теми же контрактами.
цели	Цели заявлены, и каждый шаг объявляет, что ему нужно и что он обеспечит. «Завершена» и «достигла цели» — два разных ответа об одной операции.
стратегия выбора	Кто решает, что делать дальше: встроенный алгоритм по близости к целям, своя стратегия или внешний решатель, в том числе языковая модель (D6).
поручения и ожидания	У каждого поручения наружу есть срок и правила повтора, ответ связывается с процессом. Ожидание называет, чего ждёт, и принимает команды извне. Необъявленный вид поручения наружу не уходит.
сроки и пределы	Сроки относительные — «через три дня»; календарных моментов как примитива нет. Отсчитываем их мы. Сколько раз шаг может сработать и сколько может ждать цель, задано явно; исчерпанный предел назван в записи.
обратные действия	У шага, который просто так не отменить, есть обратное действие с деловым именем. Откат — часть процесса со своими пределами и повторами; его результат проверяется правилами (D5).

Всё это — одно описание. Поэтому проверка до выпуска видит несогласованность не только внутри таблицы, но и на стыках правил, шагов, сроков и откатов (D4). Процесс не зависает молча: он заканчивается одним из двенадцати названных окончаний.

D1 Состояние процесса на выбранном такте

КАРТА ПРОЦЕССА



Какие события процесс ожидает в данный момент, какие условия уже выполнены, сколько целей достигнуто и какой срок установлен по каждому ожиданию.

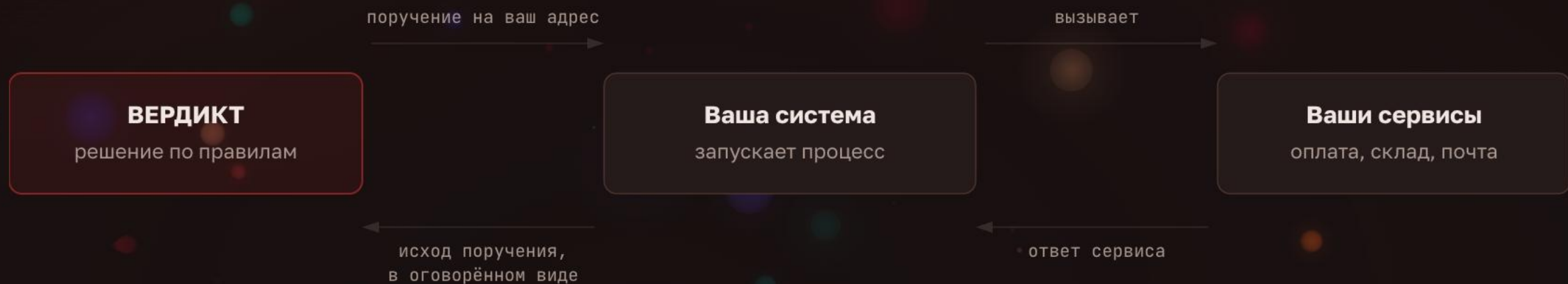
Состояние не собирается запросом по журналам: это и есть состояние сессии. Запись ведётся во время выполнения, а не восстанавливается потом.

Такт выбирается на шкале. Видно, какие данные были известны процессу на этот момент и какое решение было принято по ним.

Разбор отдельной операции не требует ни разработчика, ни выгрузки данных, ни сотрудника, помнящего историю.

Это тот же объект, что описан в D1: сессия, такты, ожидания и цели. Исполнение, проверка и запись читают одно и то же описание.

D2 Интеграционная граница и базовый интерфейс



БАЗОВЫЙ ИНТЕРФЕЙС ИНТЕГРАЦИИ

POST /api/v1/sessions	запуск процесса
GET /api/v1/sessions/{id}	состояние процесса
POST /api/v1/sessions/{id}/callback	ответ на поручение
POST /api/v1/sessions/{id}/command	сообщение извне
POST /api/v1/sessions/{id}/abort	завершить процесс

/callback и **/command** — **прикладные входы**. Они выражены в терминах предметной области: «поручение выполнено», «терминал ответил», а не «поступил HTTP-запрос».

Адреса задаёте вы: один на всё или по одному на вид поручения. Обращаемся мы только к ним.

Относительные сроки отсчитываем мы. Срок несёт каждое поручение наружу; поручение без срока не проходит разбор процесса — отказ называет поле и причину.

Владелец установки может ограничить сроки чужого процесса сверху: превышение отклоняется при публикации, и в тексте отказа названы оба значения.

D 2

Кто за что отвечает

ВСЯ ГРАНИЦА С ВНЕШНИМ МИРОМ

Фаза 1314:54:10Показать в картеПоказать в истории

НАЧАТА ТЕМ, ЧТО

внутренний такт движка

начали ждать в фазе 12

ЗАПУЩЕНО ЗДЕСЬ

Наливаем

ответ в фазе 14

ЗАПРОСИЛИ

Наливаем

ответ в фазе 14 · 55ms

Фаза 1414:54:10Показать в картеПоказать в истории

НАЧАТА ТЕМ, ЧТО

пришёл ответ Наливаем

запрос из фазы 13

ДАННЫЕ ИЗМЕНИЛИСЬ

+2 -5

показать

ОТВЕТ

Наливаем

ПОЛУЧЕН

запрос из фазы 13 · 55ms

Фаза 1514:54:10Показать в картеПоказать в истории

НАЧАТА ТЕМ, ЧТО

внутренний такт движка

начали ждать в фазе 14

ЗАПУЩЕНО ЗДЕСЬ

Закрываем резерв

ответ в фазе 17

Выдаём чек

ответ в фазе 16

ЗАПРОСИЛИ

Закрываем резерв

ответ в фазе 17 · 691ms

Выдаём чек

ответ в фазе 16 · 627ms

Фаза 1614:54:11Показать в картеПоказать в истории

НАЧАТА ТЕМ, ЧТО

пришёл ответ Выдаём чек

запрос из фазы 15

ОСТАЛОСЬ НЕЗАКРЫТЫМ

Закрываем резерв

запрос из фазы 15

ДАННЫЕ ИЗМЕНИЛИСЬ

+2

показать

Какие поручения выданы, какие получены ответы и какие события поступили извне — на одном экране, без единого адреса вашего сервиса.

Интеграция добавляет в вашу карточку одно поле — идентификатор записи.

ЧТО	ВЕРДИКТ	ВАША СИСТЕМА
запуск операции	ожидает вызова	запускает
что делать дальше	вычисляет по правилам	получает и исполняет
вызов сервисов	не выполняет	делает своими доступами
форма ответа сервиса	не видит	приводит к оговорённому
состояние операции	хранит запись и состояние	хранит предметные данные
относительные сроки	отсчитывает и возобновляет по сроку	не участвует

03 Версии и история правок

ЧТО ВОСПРОИЗВОДИТСЯ

- Версия правил фиксируется на операции. Публикация новой редакции на выполняющиеся операции не влияет.
- Опубликованная версия не действует, пока её не включили.
- Повторный запуск на тех же данных — из Конструкторской; два одинаковых прогона дают одинаковые последовательности, вплоть до идентификаторов.
- Ответ внешнего решателя приходит событием и хранится в записи.

ЧТО НЕ ВОСПРОИЗВОДИТСЯ

- Изменившиеся с тех пор внешние условия: сервис ответил иначе, курс другой, документ отозван. Это определяет человек, а не система.
- Поведение вашего обработчика, если он был изменён. Мы сохраняем возвращённый им результат, а не способ вычисления.
- Данные, которые вы не передавали процессу: решения по ним не принимались, и в записи их нет.

У описания есть история правок: каждое сохранение записано с автором и временем, а любые две редакции сравниваются деловым языком — «порог был 10 000, стал 20 000». Слом совместимости контракта данных назван словом.

Д4 Проверка до выпуска

набор случаев	Прогоняется сразу на двух версиях — прежней и новой: видно, какие случаи дали другой результат, в какой таблице и с какими значениями.
противоречие	Два правила, которые при одних и тех же входных данных требуют разного.
пробел	Входные значения, при которых не срабатывает ни одно правило.
недостижимое и лишнее	Цели, которых нельзя достичь, и шаги, которые ничего не меняют.
находка на стыке	Срок, который никто не обрабатывает; шаг обещает результат, а его логика не пишет ни одного поля, от которого тот зависит; откат, который ничего не отменяет; объявленное поручение без обработчика.
охват	Отчёт называет свой охват первой строкой, до перечня находок. Проверка отвечает на объявленный набор вопросов; догадка помечена как догадка.
чего не обещает	Найти все ошибки. Отсутствие находок не доказывает, что ошибок нет.

Новая находка при выпуске из Конструкторской не проходит молча: выпустить версию можно только явным решением, «так задумано» — с обязательной причиной, и признанный долг остаётся на виду.

05 Сбой и обратные действия

- 01 **Исход отката назван в записи, включая тот, при котором он не дошёл до конца:** это один из четырёх объявленных исходов, например «исчерпаны попытки» или «требуется человек».
- 02 **У каждого обратного действия своё деловое имя.** Удержать, списать частично, снять удержание, вернуть — разные обязательства, и в записи они различаются. Одинаковые записи вида «фиксируем оплату» не позволяют установить, что именно произошло.
- 03 **Откат — полноценная часть процесса: со своими пределами, повторами и ожиданиями.** Результат обратного действия проверяется правилами, а не берётся на веру.
- 04 **Операция может закончиться раньше своего отката: возвраты и снятия удержаний идут дальше, каждый в свой срок.** Ручного перезапуска отката нет: откат, который не дошёл до конца, кончается названным исходом.

Повторная доставка одного и того же поручения несёт один и тот же ключ `dedupe_key`; однократность действия во внешней системе обеспечивает исполнитель, не повторяя действие по этому ключу.

06 Внешний решатель, в том числе языковая модель

- 01 ИИ может участвовать в выборе следующего шага, но своих полномочий не определяет: перечень уместного считают правила **до** обращения к модели.
- 02 Перечень передаётся решателю как входные данные, а не как рекомендация. Обратиться к модели процесс может лишь тем вызовом, который в нём объявлен заранее.
- 03 Сам шаг выполняется только при сошедшихся условиях. Ответ, для которого условия не сошлись, записывается вместе с причиной отказа и не выполняется.
- 04 Ответ модели приходит событием и записывается. В записи — допустимый набор, у каждого недопустимого шага — какие условия не сошлись, что выбрала стратегия и как она называется. Молчание решателя ограничено объявленным сроком и кончается названным исходом.

ГРАНИЦА МОДУЛЯ

ВЕРДИКТ не удерживает модель — предложить она может что угодно. Работает другое: допустимое считают правила до обращения к ней, а проверку условий шага выбор отменить не может, кем бы он ни был сделан. Модель необязательна: встроенная стратегия обходится без неё. Помощника, который пишет процесс за автора, нет: авторский путь доступен программно, а проверка до выпуска одинаково строга к любому описанию.

D7 Развёртывание и данные

КАКИЕ ДАННЫЕ НУЖНЫ

- Решения принимаются по тем данным, которые вы передаёте, и работаем только с ними. Поле, не участвующее ни в одном правиле, передавать не требуется.
- Всё, с чем работает процесс, объявлено в контракте данных: чего там нет, того для процесса не существует.
- Смысл ваших данных остаётся на вашей стороне — мы в него не заходим. Запись отвечает на другой вопрос: как шло дело. Какое правило сработало, какие не подошли и по какому условию, какая редакция действовала.

ГДЕ ЭТО РАБОТАЕТ

- Ставится у вас, из архива: без исходников и без инструментов сборки — нужна только среда контейнеров.
- В поставке есть готовые проверки установки.
- Кто может открывать разбор операций, решаете вы. В Конструкторской процесс не выполняется «понарошку»: любой прогон идёт тем же путём, что и настоящая работа.
- Записи аудита можно отправлять в вашу внешнюю систему — это включается заданным адресом.

Операция, начатая на прежней сборке, доходит до конца на ней целиком, даже если новая уже принимает работу.

07 **Безопасность и эксплуатация**

- 01 Ставится у вас; снаружи открыт один порт, служебные интерфейсы слушают только локальную петлю. Наружу установка обращается только туда, куда укажете вы: поручения — на ваши адреса, описание процесса — в ваш репозиторий и только по явному действию.
- 02 Готовых секретов в поставке нет — они создаются при инициализации; сертификат входа выпускается внутри установки.
- 03 Выдавать и отзывать доступ может только владелец.
- 04 Каждая выдача и отзыв доступа записываются в журнал вместе с тем, кто действовал: человек, сервис или консоль оператора.
- 05 Подмена описания процесса обнаруживается при каждой загрузке: контрольная сумма пересчитывается и сверяется.
- 06 Обновление без простоя; резервное копирование и восстановление — в инструментах оператора.

07 Доступ людей и интеграций

- 01 Люди входят через поставщика удостоверений по стандарту OpenID Connect: учётных записей ВЕРДИКТ не хранит и токенов людям не выпускает. Встроенный поставщик — для пилота; корпоративный подключается как поставщик того же стандарта.
- 02 Доступ выдаётся четырьмя способами: по группе из токена, по домену почты, конкретному человеку и по почте того, кто ещё ни разу не входил; при первом входе доступ закрепляется за его личностью.
- 03 Роли: четыре базовые — владелец, оператор, интегратор, наблюдатель; свои объявляются конфигурацией.
- 04 Интеграция получает собственную личность: сервисный аккаунт с меткой, сроком и доступом выпускается одной операцией, токен выдаёт сама установка по гранту `client_credentials`, отзыв закрывает выдачу новых токенов.
- 05 Парольная политика, второй фактор и восстановление пароля — забота корпоративного каталога, который подключается вместо встроенного поставщика.
- 06 Выгрузка описаний процессов в репозиторий, загрузка оттуда и настройка адреса — отдельные права: ни одна роль не получает их по наследству, их выдают явно.

D8 **Гарантии**

На что можно положиться в любой операции.

- Г1 Операция приходит к концу: она либо идёт дальше, либо заканчивается названным исходом, и всегда видно, чего она ждёт.
- Г2 У каждого поручения наружу есть срок, и следит за ним сам ВЕРДИКТ. Поручение без срока не проходит разбор процесса.
- Г3 Начатая операция доходит до конца на прежней сборке, даже когда новая уже принимает работу.
- Г4 Запись восстанавливает ход дела: шаги, поручения, сроки и изменения состояния со старым и новым значением. Запись шага с ошибкой несёт саму ошибку.
- Г5 Ошибка не выдаёт себя за успех: пропущенное или испорченное обязательное значение — названная ошибка. Значений по умолчанию ВЕРДИКТ не подставляет.
- Г6 Ставится у вас из архива: без исходников и без сборки на вашей машине, нужна только среда контейнеров.
- Г7 Обновление самого ВЕРДИКТА не меняет незаметно порядок внутренних операций: такое изменение останавливает проверку перед выпуском.

Если возникает вопрос «почему не разработать это самостоятельно»

В большинстве компаний это уже разработано — в каждом сервисе по-своему, разными специалистами и в разные годы.

Именно поэтому объяснить решение, прошедшее через четыре сервиса, сегодня невозможно: в каждом из них своя структура записи, свой набор полей и своё понимание того, что считать основанием.

Сложность не в том, чтобы написать. Сложность в том, чтобы согласовать единую структуру записи между командами и сохранить её при работе в условиях сроков. Как правило, такая договорённость не выдерживает первого срочного релиза.

Промышленный пилот

Продукт готов к промышленной эксплуатации. Берём ваш процесс — тот, где меняются условия, есть сроки и цена ошибки заметна, — и доводим его до работы в вашем контуре.

Георгий Бах

george.bv@verdikt-app.ru · verdikt-app.ru