Что выполняет ВЕРДИКТ и что остаётся за вашей системой

Восемь модулей, каждый читается отдельно. Модули связаны одним примером — выплатой по рейсу SL 214: какие задания уходят вашим системам и что возвращается.

пример	Выплата Анне К. по рейсу SL 214: задания вашим системам и их ответы (с. 2)
D1	Модель исполнения: операция и шаг исполнения, контракт данных, таблицы решений, цели, выбор шага, задания, сроки, отмена сделанного (с. 3–5)
D2	Интеграционная граница: кто запускает, кто вызывает, где хранится состояние, кто отсчитывает сроки (с. 6–8)
D3	Версии и история правок: что воспроизводится и как сравниваются редакции (с. 9–10)
D4	Проверка до выпуска: случаи, противоречия, пробелы, расхождения между частями, охват (с. 10–11)
D5	Сбой и отмена сделанного: исход отмены, пределы, повторная доставка (с. 12–13)
D6	Выбор шага: перечень допустимого вычисляется до обращения к модели (с. 14–15)
D7	Развёртывание, безопасность и доступ: что остаётся в вашем контуре (с. 16–19)
D8	Надёжность: на что можно положиться (с. 20)

Пример — вымышленная выплата при задержке рейса; та же механика — например, в возвратах, доступах, закупках.

Георгий Бах

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

ПРИМЕР

Выплата Анне К. по рейсу SL 214

Рейс Санкт-Петербург — Сочи, вылет по расписанию 14:20, задержка 358 минут; программа «Базовая», условия «Лето». Какие задания ВЕРДИКТ отправляет вашим системам и что получает обратно:

ШАГ НА КАДРЕ	КОМУ ЗАДАНИЕ	ЧТО ВОЗВРАЩАЕТСЯ	МОДУЛИ
«Проверяем полис»	полисная система	программа «Базовая», условия «Лето»	D1, D2
«Узнаём статус рейса»	сервис статусов рейсов	задержка по данным сервиса: 358 минут	D2
«Оцениваем случай», «Фиксируем решение»	не уходит: считают правила ВЕРДИКТА	358 меньше 360 — первая ступень, 3 000 ₽; справка не нужна; решение «Выплатить»	D1, D3, D4, D6
«Резервируем лимит»	полисная система	исход задания: выполнен	D2, D5
«Выплачиваем»	платёжный шлюз	исход задания; итог выплаты — позже, отдельной командой «Итог выплаты»	D2, D5
«Сообщаем застрахованному»	рассылка сообщений	исход задания; выплата закрывается итогом «Выплачено»	D8

Если сумма больше 3 000 ₽, добавляется «Запрашиваем справку авиакомпании»: справку ждут 48 часов, спорное решает специалист за 24 часа (с. 15). Все задания идут на адреса, которые задали вы; D3 — версия правил закреплена за выплатой, D7 — установка у вас и доступ к записи.

Страховая компания, авиакомпания и пассажир вымышлены. Кадры в приложении — с учебного стенда: имена обезличены, сроки сокращены до секунд.

D1

Модель исполнения: что описано

сессия

То же, что операция: один проход процесса со своим состоянием, своей записью и своей версией правил — той, на которой операция началась. В примере — выплата Анне К. по рейсу SL 214: одна операция от проверки полиса до итога.

такт

Один цикл работы (на кадрах подписан «фаза»): процесс возобновляется, смотрит на текущие данные, вычисляет по ним правила и выбирает следующий шаг из допустимых, а не идёт по заранее нарисованному маршруту. В примере: пришёл ответ о статусе рейса — выбран шаг «Оцениваем случай».

контракт данных

Схема данных процесса: состояние, факты, задания вашим системам. Все данные описаны схемой, поля вне схемы процессу недоступны. В примере: полис, рейс, задержка в минутах, сумма к выплате.

таблицы решений

Решение раскладывается на читаемые ступени, таблицы ссылаются друг на друга. Условие — фраза, а не код, и его можно разобрать формально. Логика шагов пишется рядом, по той же схеме данных. В примере: «Границы ступеней задержки» (180, 360 и 720 минут) и «Диапазон задержки по данным сервиса».

цели

У процесса заданы цели. У каждого шага указано, что ему нужно и что он даёт. «Завершена» и «достигла цели» — два разных ответа об одной операции. В примере: шагу «Оцениваем случай» нужны данные полиса и закрытый вопрос о статусе рейса, он даёт «Случай оценён».

D1

Модель исполнения: как исполняется

кто выбирает шаг

Следующий шаг выбирает встроенный алгоритм по близости к целям, ваш алгоритм либо сотрудник, сторонняя система или языковая модель (D6). Модель необязательна. В примере допустимые шаги, выбранный и условия остальных видны в записи (с. 14–15).

задания и ожидания

У каждого задания вашим системам есть срок и правила повтора, ответ связывается с процессом. Для каждого ожидания указано, чего оно ждёт; команды извне принимаются и во время ожидания. Задание вида, которого нет в описании процесса, вашим системам не отправляется. В примере: «Проверяем полис» уходит в полисную систему, «Выплачиваем» — в платёжный шлюз; пока выплата ждёт, приходит команда «Итог выплаты» (с. 7–8).

сроки и пределы

Сроки относительные — «через три дня»; сроков вида «в 9:00» нет. Отсчитываем их мы. Сколько раз шаг может сработать и сколько может ждать цель, задано явно; исчерпанный предел назван в записи. В примере: справку авиакомпании ждут 48 часов, ответа специалиста — 24 часа.

отмена сделанного

У шага, который просто так не отменить, есть своя отмена с деловым именем (на кадрах — «откат»). Отмена — часть процесса со своими пределами и повторами; её результат проверяют правила (D5). В примере: резерв лимита снимается отдельным шагом «Снимаем резерв лимита» (с. 13).

окончание

Процесс не повисает молча: он заканчивается одним из заранее названных окончаний. Окончания — общий перечень платформы, исходы — названия результата конкретной операции. В примере: «Выплачено», «Выплачена бесспорная часть, остаток у руководителя».

Всё это — одно описание. Поэтому проверка до выпуска видит несогласованность не только внутри таблицы, но и в расхождениях между правилами, шагами, сроками и отменой сделанного (D4).

D1

Состояние выплаты на выбранном шаге

ВЫПЛАТА ПО РЕЙСУ SL 214

ПОЛИС

Программа: Базовая

Условия: Условия «Лето»

РЕЙС

Рейс: SL 214

Маршрут: Санкт-Петербург — Сочи

Плановый вылет: 01.10.2026, 14:20

Тип рейса: Регулярный

Задержка (сервис), мин: 358 мин

ОЦЕНКА

Диапазон (сервис): Первая ступень

Сумма по данным сервиса: 3 000 ₽

РЕШЕНИЕ

Решение: Выплатить

К выплате: 3 000 ₽

На чём решение: По данным сервиса

Рейс SL 214, 358 минут, первая ступень, «Выплатить», 3 000 ₽.

Видно, чего процесс ждёт сейчас, какие условия выполнены, какие цели достигнуты и какой срок установлен по каждому ожиданию.

Состояние хранится как есть, его не восстанавливают по журналам. Запись ведётся во время выполнения, а не собирается потом.

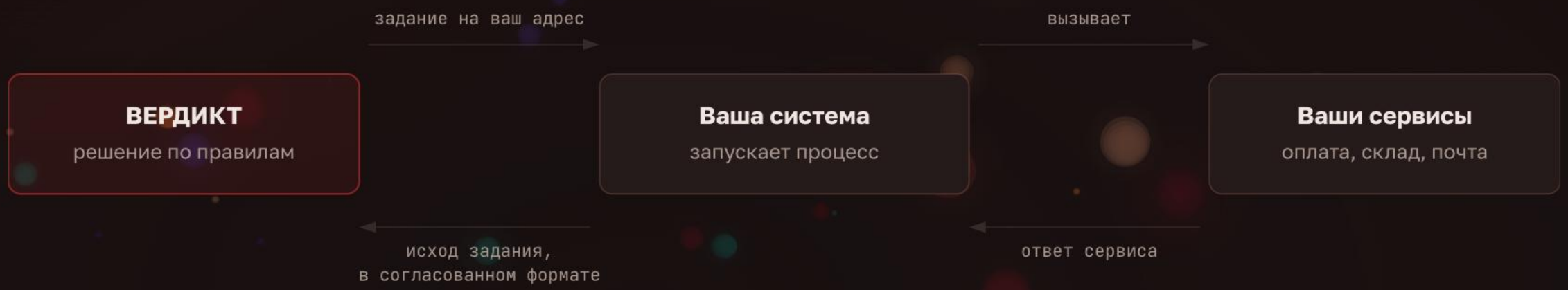
Состояние открывается на выбранном шаге исполнения: видно, какие данные были известны процессу на этот момент и какое решение принято по ним.

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

Это тот же объект, что описан в D1: операция, шаги исполнения, 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 задание
`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: «задание выполнено», «терминал ответил», а не «поступил HTTP-запрос».

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

Относительные сроки отсчитываем мы. Срок есть у каждого задания вашим системам; задание 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.

D2

Образец обмена: задание «Выплачиваем»

01 **ВЕРДИКТ отправляет задание на ваш адрес.** Задание шага «Выплачиваем» несёт срок и правила повтора из описания процесса и ключ `dedupe_key`. Адрес задаёте вы.

02 **Ваша система вызывает платёжный шлюз сама,** своими доступами. ВЕРДИКТ к шлюзу не обращается и формы его ответа не видит.

03 **Ваша система отвечает на задание:** `POST /api/v1/sessions/{id}/callback`. Из ответа шлюза она берёт то, что нужно для следующего решения, и возвращает исход задания в согласованном формате.

04 **Итог выплаты приходит позже,** отдельным сообщением: `POST /api/v1/sessions/{id}/command`. Процесс принимает его и во время ожидания, ход операции меняется сразу.

05 **Если выплата выполнена, а ответ потерялся,** задание придёт повторно с тем же ключом `dedupe_key`. По нему ваша система не повторяет выплату: однократность действия обеспечивает она.

ВЕРДИКТ

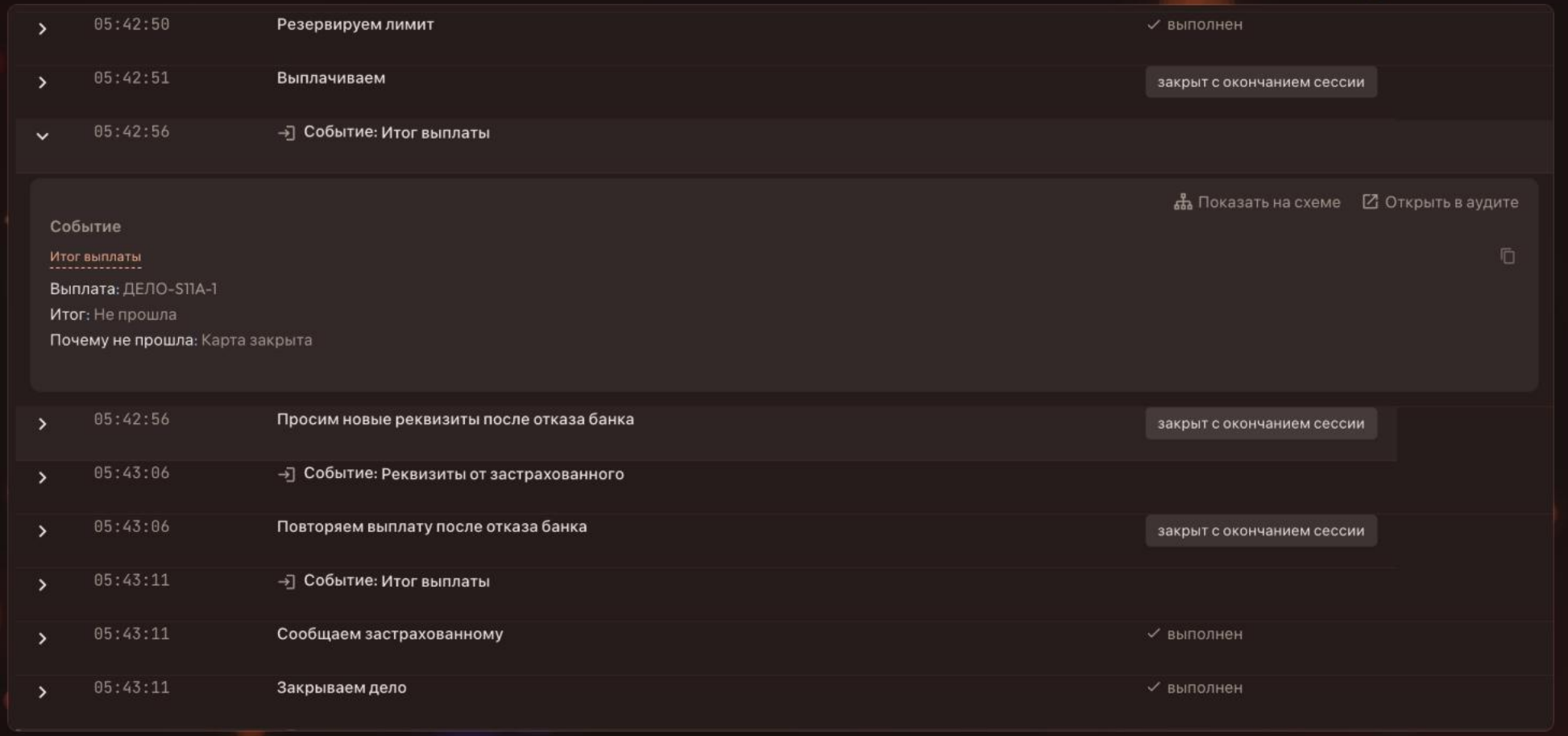
- Ожидает вызова, вычисляет по правилам, что делать дальше.
- Хранит запись и состояние операции.
- Отсчитывает относительные сроки и возобновляет работу по сроку.

ВАША СИСТЕМА

- Запускает операцию, получает и исполняет задания.
- Вызывает сервисы своими доступами, приводит их ответ к согласованному формату.
- Хранит предметные данные; в карточку добавляется одно поле — идентификатор записи.

D 2

Ход выплаты: задания и команды извне



Задания «Резервируем лимит», «Выплачиваем», «Просим новые реквизиты», «Повторяем выплату»; команды извне «Итог выплаты» (банк: карта закрыта) и «Реквизиты от застрахованного»; затем сообщение застрахованному и закрытие.

D3

Версии и история правок

ЧТО ВОСПРОИЗВОДИТСЯ

- Версия правил фиксируется на операции. Публикация новой редакции на выполняющиеся операции не влияет.
- Выпущенная версия не действует, пока её не включили.
- Повторный запуск на тех же данных — из конструктора (там правила пишут и проверяют); два прогона дают один и тот же результат, вплоть до идентификаторов.
- Ответ того, кто выбирает шаг, 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 сервиса, принимающего задания, если его изменили. 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.

ВЫПЛАТА X: НАЧАТА ДО ВКЛЮЧЕНИЯ НОВОЙ ВЕРСИИ

Задержка 2 ч 30 мин. Завершается по прежним правилам: «выплата не положена».

ВЫПЛАТА Y: ТАКАЯ ЖЕ, НАЧАТА ПОСЛЕ

Идёт по новым правилам: первая ступень, 3 000 ₽.

У описания есть история правок: каждое сохранение записано с автором и временем, а любые две редакции сравниваются деловым языком — «порог был 180 минут, стал 120». Несовместимое изменение контракта данных помечается явно.

D3 · D4

Изменение порога: одна ячейка и четыре примера

ТАБЛИЦА «ГРАНИЦЫ СТУПЕНЕЙ ЗАДЕРЖКИ»

Границы ступеней задержки

Отвечает Границы ступеней

#	ЕСЛИ	ТО
1	во всех остальных случаях	*Единственное место, где записаны часы: 3 ч, 6 ч, 12 ч (Рег. п. 1, 2). Правка порога — одна ячейка* Начало первой ступени **180** Начало второй ступени **360** Начало третьей ступени **720**

Часы записаны в одном месте: 180, 360 и 720 минут. Порог снижают с 180 до 120 минут — правится одна ячейка.

01 **Сначала записываем, как должно быть:** «120 мин → первая ступень» и «179 мин → первая ступень». Ступень определяют две таблицы — по данным сервиса статусов рейсов и по справке авиакомпании, — поэтому на прежних правилах не проходят ровно четыре примера.

02 **Меняем число.** Набор прогоняется сразу на двух версиях — прежней и новой: проходят все примеры, остальные не изменились, видно, в какой таблице и с какими значениями.

03 **Новые правила включает ответственный человек.** Выпущенная версия не действует, пока её не включили; программисты и выпуск прикладного кода не нужны.

D4

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 (с. 10).
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.
расхождение между частями	Правила, шаги, сроки и действия при сбое не согласованы друг с другом: срок, который никто не обрабатывает; шаг обещает результат, а его логика не пишет ни одного поля, от которого тот зависит; отмена, которая ничего не отменяет; задание без обработчика.
охват	Отчёт называет свой охват первой строкой, до перечня находок. Проверка отвечает на заранее описанный набор вопросов, а догадка в отчёте помечена как догадка.
чего не обещает	Найти все ошибки. Отсутствие находок не доказывает, что ошибок нет.
выпуск и включение	Выпуск — новая версия прошла проверку и опубликована из конструктора. Включение — человек делает выпущенную версию действующей, и новые операции идут по ней.

Новая находка при выпуске из конструктора не проходит молча: выпустить версию можно только явным решением, «так задумано» — с обязательной причиной; принятые так находки остаются на виду счётом и не исчезают.

D5

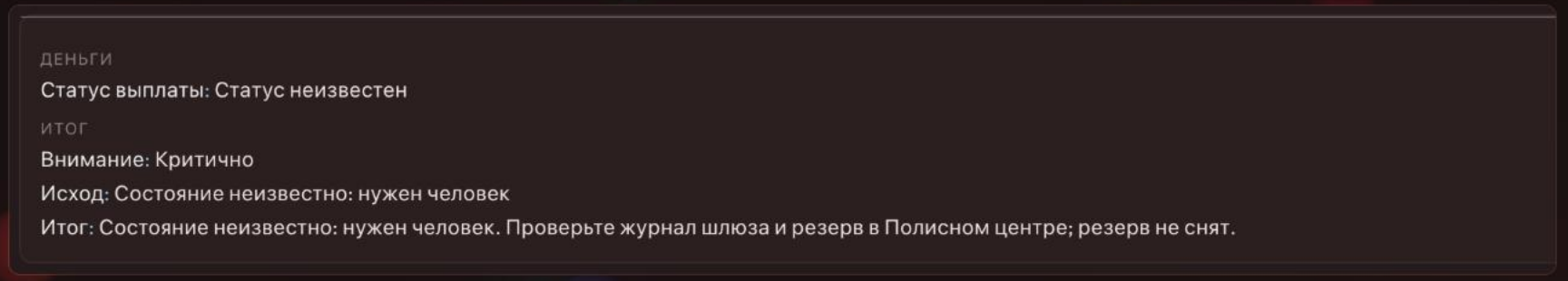
Сбой и отмена сделанного

01 **Чем закончилась отмена, видно в записи, в том числе если она не дошла до конца:** это один из четырёх заранее названных исходов, например «исчерпаны попытки» или «требуется человек».

02 **У каждой отмены 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 **Отмена — полноценная часть процесса: со своими пределами, повторами и ожиданиями.** Результат отмены проверяется правилами, а не берётся на веру.

04 **Операция может закончиться раньше своей отмены:** возвраты и снятия удержаний идут дальше, каждый в свой срок. Ручного перезапуска отмены нет: если отмена не дошла до конца, это прямо записано.



Шлюз принял выплату, итог в срок не пришёл: резерв не снят, в итоге указано, что проверить. Исход — «Состояние неизвестно: нужен человек».

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

D5

Шлюз отказал: сделанное отменено

Сделанное сессией отменено
Не удался шаг: Выплачиваем

Итог

Статус выплаты	Внимание	Исход
Шлюз не принял поручение	Критично	Откат выполнен, выплаты нет

Что случилось

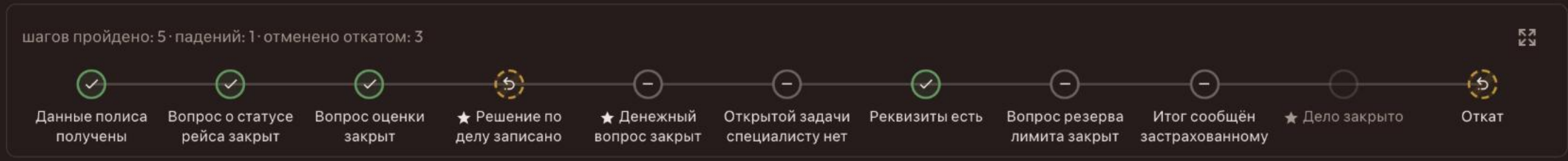
- Не удался шаг: Выплачиваем
- Откат начат: первым отменяется «Резервируем лимит»
- Откат закончен: Оцениваем случай

Последствия

Отменено

- Резервируем лимит
- Фиксируем решение
- Оцениваем случай

Схема процесса



Шлюз не принял выплату — ВЕРДИКТ отменил сделанное и назвал итог: выплаты нет (на кадре «откат» — отмена сделанного).

D6

Кто выбирает шаг, 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 тому, кто выбирает шаг, как входные данные, а не как рекомендация. Обратиться к модели процесс может лишь тем вызовом, который в нём описан 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 выбрано и как называется способ выбора. Молчание того, кто выбирает шаг, ограничено заданным сроком, и чем оно закончилось, видно в записи.

ГРАНИЦА МОДУЛЯ

ВЕРДИКТ не ограничивает, что может предложить модель: ограничения задают правила. Допустимое считают правила до обращения к ней, а проверку условий шага выбор отменить не может, кем бы он ни был сделан. Модель необязательна: встроенный выбор обходится без неё. Помощника, который пишет процесс за автора, нет: авторский путь доступен программно, а проверка до выпуска одинаково строга к любому описанию.

D6

Если специалист не ответил в срок

ВЫПЛАТА С СУММОЙ БОЛЬШЕ 3 000 ₽

СПРАВКА

Справка перевозчика: Не пришла в срок

ОЦЕНКА

Сумма по данным сервиса: 6 000 ₽

Бесспорно: 3 000 ₽

СПЕЦИАЛИСТ

Решение специалиста: Не ответил в срок

Тема задачи: Справка не пришла при крупной сумме

РЕШЕНИЕ

К выплате: 3 000 ₽

На чём решение: Бесспорная часть

ДЕНЬГИ

Выплачено: 3 000 ₽

Спорная часть: 3 000 ₽

ИТОГ

Исход: Выплачена бесспорная часть, остаток у руководителя

Справка не пришла в срок, по данным сервиса 6 000 ₽, бесспорно 3 000 ₽, специалист не ответил. К выплате 3 000 ₽.

Без справки авиакомпании платят до 3 000 ₽. Больше — только по справке, её ждут 48 часов. Спорное решает специалист за 24 часа.

Если специалист молчит, ожидание ограничено заданным сроком и кончается названным исходом: выплачена бесспорная часть, остаток у руководителя группы.

Кто выбирает шаг — сотрудник, ваша система или модель, — допустимое перед этим вычисляют правила. Модель необязательна.

Итог на кадре — «Выплачена бесспорная часть, остаток у руководителя». Срок справки и срок ответа специалиста записаны в описании процесса.

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.
- Все данные процесса описаны схемой — контрактом данных. Поля вне схемы процессу недоступны.
- Смысл ваших данных остаётся на вашей стороне — мы в него не заходим. Запись отвечает на другой вопрос: как шла операция. 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 конструкторе процесс не имитируется: любой прогон идёт тем же путём, что и рабочая операция.
- 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.

D7

Состав, поставка и эксплуатация

исполнение	Падение узла посреди операции не оставляет процесс в неизвестном состоянии: исполнение восстанавливается по записи.
поставка и установка	Архив: готовые исполняемые файлы, собранный интерфейс, конфигурация, книга оператора, приёмочный набор и образы контейнеров. Нужна только среда контейнеров. Одна команда инициализации создаёт секреты, заводит первого администратора и проверяет вход по защищённому каналу.
сеть и сертификат	Наружу открыт один порт: за ним конструктор, рабочее место владельца процесса и аудитора, управление доступом, вход людей и выдача токенов интеграциям. Служебные интерфейсы слушают только локальную петлю. Сертификат входа выпускается внутри установки, без внешних центров сертификации.
секреты	Хранилище секретов в одном из трёх режимов: своё, распечатывающее себя само; своё в режиме долей, когда распечатать его может только кворум держателей; хранилище заказчика. Мастер-ключи поворачиваются отдельными командами.
целостность	Целостность каждого файла и образа архива сверяется с манифестом до установки; подмена отказывает с названием файла. Сборка выпуска воспроизводима до байта.
обновление	Без простоя: новая версия поднимается рядом, вход переключается, старая доигрывает свои операции. Перед изменением установки печатаются план и цена; обрыв доводится повтором той же команды.
журнал доступа	Уходит в вашу систему сбора событий по syslog: UDP, TCP или TLS, форматы RFC 5424 или CEF. Отправка идёт через очередь: при обрыве связи записи не теряются. Выключить журнал настройками нельзя.
изоляция	Читатель из чужого арендатора получает ответ «не найдено» — такой же, как для несуществующего идентификатора.

Это свойства реализации, а не перечень сертификаций.

D7

Безопасность и эксплуатация

01 Ставится у вас; подключение к установке — через один сетевой порт в вашей сети, служебные интерфейсы доступны только на самом сервере. Сама установка обращается только к адресам, которые укажете вы: задания — 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; программа оператора проверяет совместимость выпуска и права до первого изменения установки.

D7

Доступ людей и интеграций

01 Люди входят через систему единого входа по стандарту OpenID Connect: учётных записей ВЕРДИКТ не хранит и токенов людям не выпускает. Встроенная система входа — для пилота; корпоративная подключается по тому же стандарту.

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 учётной записью.

03 Роли: четыре базовые — владелец, оператор, интегратор, наблюдатель; свои задаются конфигурацией.

04 Интеграция получает собственную учётную запись — 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 встроенной системы входа.

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

Надёжность

На что можно положиться в любой операции (Г1–Г7 — постоянные номера свойств).

Г1 Операция всегда приходит к концу: либо идёт дальше, либо заканчивается с понятным итогом — и всегда видно, чего она ждёт.

Г2 У каждого задания вашим системам есть срок, и следит за ним сам ВЕРДИКТ. Задание 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 операции: шаги, задания, 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