Кто решает, а кто выполняет

Агент, модель, обвязка и среда исполнения

1 **Агент**
Вся программа: получает задачу и ведёт работу до результата.

2 **Модель · LLM**
По инструкциям и данным предлагает ответ или следующее действие.

3 **Обвязка и среда исполнения**
Harness управляет циклом и вызовами. Runtime — среда, где работает код.

В СВОДКЕ SOC

Модель предлагает уточнить историю заявки. Обвязка проверяет запрос, запускает инструмент и возвращает ответ модели.

Предложение модели ещё не означает разрешение на действие.

Инструкция, операция, модуль

Скилл, инструмент и расширение

1. **Скилл · skill**
   Инструкции и материалы: как выполнить определённую задачу.

2. **Инструмент · tool**
   Конкретная операция: прочитать заявку или сохранить черновик.

3. **Расширение · plugin**
   Добавляет возможности приложению; состав зависит от платформы.

В СВОДКЕ SOC

Скилл требует проверить историю заявки. Инструмент читает её. Расширение может поставлять оба компонента.

Инструкция объясняет порядок работы. Права доступа выдают отдельно.

Запрос ещё не результат

Инструмент, его вызов и ответ

1 **Инструмент**
get_incident — доступная операция чтения карточки.

2 **Вызов · tool call**
Прочитать INC-041 — запрос с конкретным номером инцидента.

3 **Результат · tool result**
Данные карточки, подтверждение действия или сообщение об ошибке.

В СВОДКЕ SOC

Модель запросила INC-041. Проверка прав отклонила чтение. В журнале есть вызов, но данные агент не получил.

Различайте: запросили → разрешили → выполнили.

Как добраться до данных

MCP, API и эндпоинт

1 **MCP**
Протокол связи приложения агента с сервером инструментов и данных.

2 **API**
Программный интерфейс системы, например SIEM.

3 **Эндпоинт · endpoint**
Конкретная точка обращения к сервису, например адрес карточки.

В СВОДКЕ SOC

Приложение → по MCP к серверу инструментов → через API к SIEM. Возможна и прямая интеграция, без MCP.

Подключение по MCP само по себе не делает доступ безопасным.

Что агент помнит сейчас

Сессия, контекст и память

1. **Сессия**
   История работы, которую объединяет и хранит приложение.

2. **Контекст**
   Инструкции и данные, переданные модели в конкретном запросе.

3. **Долговременная память**
   Сведения, сохранённые для последующих обращений и запусков.

В СВОДКЕ SOC

Формат сводки можно сохранить в памяти. Состояние инцидентов надо запросить заново. В контекст попадёт выбранная часть данных.

Приложение может хранить больше, чем модель видит в этом запросе.

Кто разрешил это действие

Инструкция, учётная запись и права

1 **Инструкция**
Что поручено сделать: подготовить утреннюю сводку.

2 **Учётная запись · identity**
От чьего имени программа обращается к корпоративной системе.

3 **Права доступа**
Какие операции с какими данными разрешает эта система.

В СВОДКЕ SOC

Учётной записи soc-briefing запрещено удалять заявки. Запрет должен действовать и через API, и при нажатии кнопки в браузере.

Общий инструмент клика тоже может удалить заявку. Права проверяет сама система.

Один вызов — несколько связей

Что происходит внутри инструмента

1 **Агент вызывает инструмент**
get_incident: получить сведения об INC-041.

2 **Инструмент обращается к системам**
SIEM — за карточкой, система заявок — за историей, сервис доступа — за токеном.

3 **Модель получает результат**
Приложение отдельно передаёт выбранные данные сервису модели.

В СВОДКЕ SOC

Локальный MCP-сервер может общаться с приложением без сети, а сам выполнять запросы в интернет.

Список инструментов не заменяет список фактических сетевых обращений.

Откуда взялась строка отчёта

Источник данных и журнал действий

1 **Источник · provenance**
IT-118: пароль сменили. Подтверждения завершения сеансов нет.

2 **Действия · trace**
Запуск → запрос IT-118 → проверка прав → ответ с отметкой времени.

3 **Вывод в сводке**
Нужно проверить завершение сеансов. Прекращение доступа ещё не подтверждено.

В СВОДКЕ SOC

Сохраните номер и версию записи, время получения и общий номер запуска. Завтра комментарий в заявке могут изменить.

У каждого существенного вывода должен быть проверяемый источник.