Вебинар

ELMA365

elma365.ru

Топ-5 вызовов для CIO/CDTO

Баланс между технологическим долгом и темпом цифровизации

Андрей Чепакин

Вице-президент по стратегическим продажам и партнерствам ELMA

Топ-5 вызовов для CIO / CDTO

1

Баланс между технологическим долгом и темпом цифровизации

2

Переход от «зоопарка» систем к цифровой платформе

3

AI/ML: от пилотов к масштабируемой ценности

4

Роль CIO: от подрядчика к стратегическому партнеру

5

Удержание и развитие ИТ-команды

Баланс между техдолгом и темпом цифровизации

Потребность бизнеса:

Быстро выводить на рынок продукты и решения, масштабировать сервисы, опережать конкурентов.

Как формулирует потребность бизнес:

«Сделайте это быстро» «Завтра должно работать» «Нам нужно MVP к следующему кварталу» «Нам нужная поддержка стратегии компании со стороны ИТ»

Почему это сложность для CIO:

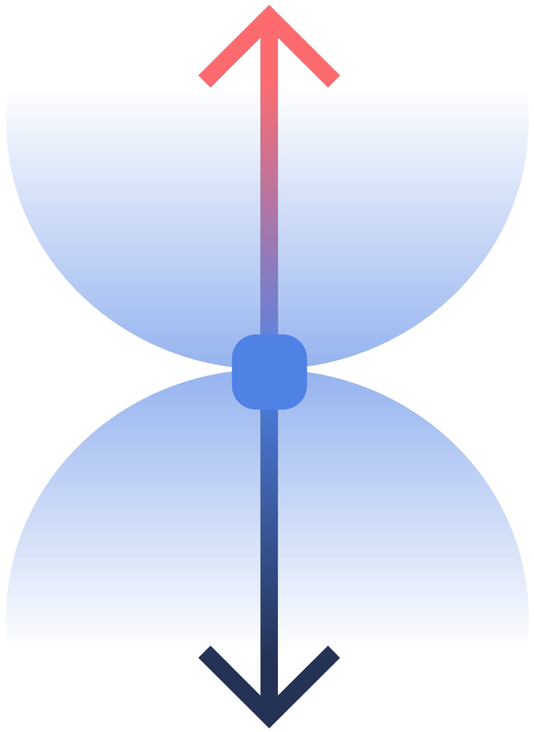
Скорость достигается за счёт компромиссов: рост техдолга, нестабильная архитектура, сложная поддержка.

В долгосрочной перспективе — это снижает гибкость и повышает издержки.

Варианты действий:

- Игнорировать долг и «гасить пожары»
- Строить системную работу по инвентаризации и контролю техдолга
- Увязывать развитие с архитектурной стратегией

Стратегический уровень: перспектива компании



СУТЬ ВЫЗОВА

Стратегическая цель компании — быть **конкурентоспособной** за счёт цифровой скорости, гибкости, клиентского опыта

Но хаотичная цифровизация создаёт эффект замедления: фрагментированная архитектура, неинтегрированные решения, технический долг, зависимость от ключевых людей.

СТРАТЕГИЧЕСКИЕ РИСКИ

- Потеря управляемости цифровыми активами
- Рост OpEx из-за высокой стоимости поддержки старых решений
- Потеря времени на интеграции и доработки → ухудшение Time-to-Market

Потребность бизнеса

Быстро выводить на рынок цифровые продукты и решения, масштабировать сервисы, опережать конкурентов.

Стратегическая цель компании —

БЫТЬ КОНКУРЕНТОСПОСОБНОЙ

за счёт цифровой скорости, гибкости и клиентского опыта

Конкурентоспособность — это способность компании:

- удерживать и наращивать долю рынка,
- обеспечивать прибыльность,
- адаптироваться к изменениям быстрее и/или эффективнее конкурентов.

Это не статичное преимущество, а динамическая способность быть "на шаг впереди" в условиях неопределенности: рынка, технологий, потребителей, регуляторов.

СКОРОСТЬ = TIME-TO-MARKET

Быстрее вывел продукт → раньше получил обратную связь → быстрее улучшил → обогнал конкурента

- Технологический цикл сужается.
- Быстрые компании первые занимают нишу, задают стандарты.
- Скорость ≠ хаос. Это про сокращение цикла Идея → Ценность.

ГИБКОСТЬ = СПОСОБНОСТЬ АДАПТИРОВАТЬСЯ

Мир меняется — побеждает не сильный, а тот, кто быстрее подстраивается.

- Быстрая смена приоритетов, сценариев, бизнес-моделей.
- Технологии, процессы и кадры должны быть адаптивны.
- Гибкость = не только в IT, но и в культуре, управлении, финансах.

КЛИЕНТСКИЙ ОПЫТ = УДЕРЖАНИЕ И РОСТ

Продуктов много. Выбирают не только по цене, а по удобству, эмоции, вовлечённости.

- Digital-first → ценится UX выше бренда.
- Компании с лучшим опытом растут быстрее и реже теряют клиентов.
- Цифровой CX — это микс: каналы, данные, персонализация, скорость, поддержка.

ИННОВАЦИИ

Устойчивое отличие от конкурентов, выход за рамки конкуренции

ОПЕРАЦИОННАЯ ЭФФЕКТИВНОСТЬ

Себестоимость, маржинальность, скорость поставки

ЭКОСИСТЕМНОСТЬ

Формирование сети партнёров/платформ вокруг себя

КАДРЫ И КУЛЬТУРА

Люди реализуют стратегии. Культура — проводник скорости и гибкости

ДОСТУП К ДАННЫМ И АНАЛИТИКЕ

Быстрая реакция на изменения, управление поведением клиента

Сложность для ИТ

Скорость достигается за счёт компромиссов: рост техдолга, нестабильная архитектура, сложная поддержка. В долгосрочной перспективе — это снижает гибкость и повышает издержки.

Хаотичная цифровизация

СОЗДАЁТ ЭФФЕКТ ЗАМЕДЛЕНИЯ:

фрагментированная архитектура, неинтегрированные решения, технический долг, зависимость от ключевых людей.

РОСТ ТЕХНОЛОГИЧЕСКОГО ДОЛГА → СНИЖЕНИЕ ГИБКОСТИ

Когда решения принимаются «на лету», без учёта архитектуры, без фиксации решений, без оценки последствий — накапливается техдолг:

- Разнородные технологии (зоопарк)
- Костыльные интеграции
- Дублирование функций и данных

ФРАГМЕНТИРОВАННАЯ АРХИТЕКТУРА → СЛОЖНАЯ ПОДДЕРЖКА И МАСШТАБИРОВАНИЕ

Хаотичная цифровизация рождает «архитектурные острова»: каждое новое решение живёт своей жизнью. Это порождает:

- Избыточную сложность
- Проблемы интеграции
- Зависимость от отдельных экспертов

ПОТЕРЯ УПРАВЛЯЕМОСТИ И ПРОЗРАЧНОСТИ

Когда цифровизация идёт без единой модели управления:

- решения внедряются без общей карты,
- сложно понять, что у нас есть, в каком оно состоянии и кто за это отвечает.

Это тормозит: планирование, управление рисками, принятие обоснованных решений на уровне руководства.

ОТСУТСТВИЕ ПЕРЕ-ИСПОЛЬЗОВАНИЯ И СИНЕРГИИ

Хаотичная цифровизация = каждый продукт «сам за себя». Это приводит к:

- повторной разработке типовых компонентов,
- низкой переиспользуемости платформенных решений,
- потере эффекта масштаба.

РОСТ ЗАТРАТ НА СОПРОВОЖДЕНИЕ И ИНТЕГРАЦИИ

Быстрые MVP без архитектурных стандартов → много ручного труда и "переизобретения велосипеда". В итоге:

- OpEx растёт из-за поддержки хрупкой инфраструктуры,
- появляется множество одноразовых решений,
- интеграции становятся трудозатратными.

Задача CIO

CIO должен построить модель, в которой цифровая скорость достигается не вопреки архитектуре, а за счёт неё.

Таким образом, увязать темпы цифровизации с архитектурной устойчивостью, чтобы не потерять гибкость, масштабируемость и управляемость ИТ-ландшафта.

Стратегический ответ

Формализовать архитектурное развитие как часть стратегии

Включить принципы эволюции ИТ-ландшафта в цифровую повестку

Ввести KPI по архитектурной устойчивости

Например:
процент сервисов, соответствующих целевой архитектуре, или Legacy Ratio по ключевым системам

Техдолг как управляемый актив

Он должен быть видимым на уровне C-level и инвестиционных комитетов

Внедрение Digital Architecture Framework

Цель:

Построить прозрачную и устойчивую целевую архитектуру, которая поддерживает масштабируемую цифровизацию.

Задачи:

1. Определить принципы архитектуры и стандартов разработки.
2. Создать карту цифровых активов и сервисов.
3. Обеспечить архитектурную согласованность между командами.

План действий:

1. Провести аудит текущей архитектуры.
2. Разработать и утвердить целевую модель (технологии, интеграции, правила).
3. Завести цифровой реестр компонентов (API, сервисы, платформы).
4. Создать архитектурные гайдлайны и шаблоны.

Результат:

1. Единая архитектурная модель.
2. Повышение переиспользования.
3. Меньше разрозненных решений

Внедрение Digital Architecture Framework

Архитектурные принципы

Reuse first (Переиспользование в приоритете) «Не создавать то, что уже есть»	API-centric (Приоритет API) «Системы общаются через открытые API, а не прямые интеграции»	Cloud-native (Приоритет облачных решений) «Приоритет решениям, масштабируемым в облаке»	Security by design «Безопасность на уровне архитектурных решений»	Data-driven (Ориентация на данные) «Данные — ключевой актив и точка принятия решений»

Целевая архитектура (Reference Architecture)

Уровень интерфейсов (Experience Layer): ▪ Web и мобильные интерфейсы (SPA, Mobile Apps) ▪ Low-code приложения	**Уровень микросервисов и API (API Layer):** ▪ RESTful API, GraphQL ▪ API Gateway (например, Kong, Apigee)	**Уровень платформы приложений (Application Platform Layer):** ▪ Low-code платформа (ELMA365) ▪ Контейнеризация и оркестрация (Kubernetes, Docker)	**Интеграционный слой (Integration Layer):** ▪ ESB или Kafka для потоковой интеграции ▪ Message Brokers (RabbitMQ, Apache Kafka	**Уровень данных и аналитики (Data & Analytics Layer):** ▪ Data Lake / Warehouse (ClickHouse, Snowflake, BigQuery) ▪ BI & ML платформы (Power BI, Tableau, MLflow)

Архитектурные стандарты и требования

- Форматы интеграций: REST API, JSON, gRPC
- Стандарты документации API: OpenAPI (Swagger)
- Единые стандарты безопасности (OAuth 2.0, JWT)
- CI/CD-практики: GitLab, GitHub Actions
- Observability-платформа: Prometheus, Grafana, Jaeger

Архитектурные инструменты

- Архитектурный инвентарь: LeanIX, Ardoq
- Управление ADR: Confluence, Git, Log4Brains
- Моделирование архитектуры: ArchiMate, Draw.io
- CI/CD и контроль качества: GitLab CI/CD, SonarQube
- Мониторинг и observability: Prometheus, Grafana, Elastic

Процессы управления архитектурой

Процесс принятия решений (ADR):
- Контекст → Решение → Альтернативы → Последствия → Статус
- Единое хранилище ADR с обязательным review

Архитектурные ревью:
- Архитектурные гильдии — регулярные встречи для обсуждения решений и ADR
- Архитектурное ревью обязательная часть релизного процесса

Процесс работы с техдолгом:
- Техдолг-реестр с критериями приоритизации
- Включение задач по устранению техдолга в квартальные планирования
- Регулярная отчётность по техдолгу на уровне CIO/CDTO

Метрики архитектурной устойчивости

Legacy Ratio	Доля legacy-компонентов в архитектуре	Снижение < 20%
ADR Coverage	% решений, оформленных как ADR	> 90%
Change Lead Time	Время от идеи до внедрения	Снижение на 20% ежегодно
API Reuse Ratio	Уровень переиспользования API	Повышение на 10–15% ежегодно

Роли и ответственность

CIO/CDTO	Архитектурная стратегия, KPI, инвестиции
Chief Architect	Целевая архитектура, ADR, архитектурные стандарты
Product Owner	Применение архитектурных решений, ADR
Архитекторы в командах	Соблюдение стандартов и ADR, участие в planning и ревью
DevOps-команды	Внедрение и поддержка CI/CD, Observability
Архитектурные гильдии	Обсуждение, ревью, развитие архитектурных практик

Построение карты технологического долга

Цель:

Перевести техдолг из «невидимого шума» в управляемую часть цифровой повестки.

Задачи:

1. Визуализировать техдолг.
2. Приоритизировать критические зоны.
3. Интегрировать его в backlog и квартальное планирование.

План действий:

1. Построить реестр техдолга по системам/продуктам.
2. Утвердить критерии приоритизации (влияние, риск, частота изменений).
3. Создать шаблоны задач техдолга в Jira.
4. Ввести регулярные ревью на уровне команд и портфеля.

Результат:

1. Видимость техдолга на уровне C-level
2. Планомерное снижение риска и затрат на поддержку.
3. Включение арх. улучшений в рабочую повестку команд.

Построение карты технологического долга

Шаг процесса	Описание	Участники	Результат
Идентификация и регистрация техдолга	▪ Сбор данных о потенциальном техдолге (от команд разработки, поддержки, архитектуры). ▪ Регистрация в едином реестре техдолга (например, Jira, Confluence).	Архитекторы, команды разработки, Product Owners.	Заполненный и актуальный реестр техдолга с базовыми метриками (критичность, влияние, причина).

Классификация и приоритизация	▪ Определение категорий техдолга (архитектурный, кодовый, инфраструктурный, процессный).
	▪ Приоритизация техдолга по критериям:
	  ▪ Влияние на бизнес (время релизов, стабильность, затраты).
	  ▪ Сложность и стоимость устранения.
	  ▪ Риск невмешательства.
		Chief Architect, Архитектурные гильдии, Product Owners.	Актуализированный реестр с расставленными приоритетами (high, medium, low).

Планирование работ с техдолгом	▪ Включение задач по устранению техдолга в backlog команд на ежеквартальном планировании. ▪ Формирование бюджета (Refactoring budget) и согласование с C-level.	Product Owners, CIO/CDTO, Архитекторы, PMO.	Формализованный план задач по устранению техдолга, включённый в дорожные карты и backlog.
Реализация и контроль устранения	▪ Выполнение задач по устранению техдолга согласно приоритетам в backlog. ▪ Архитектурное и техническое ревью выполненных задач. ▪ Мониторинг прогресса и регулярный апдейт статуса устранения.	Команды разработки и поддержки, Архитекторы.	Постепенное снижение общего уровня техдолга и достижение целевых KPI.
Мониторинг и анализ прогресса	▪ Регулярная отчётность (например, ежемесячно, ежеквартально) по уровню техдолга, KPI и достигнутым результатам. ▪ Оценка эффективности процесса управления техдолгом (метрики: Legacy Ratio, Tech Debt Remediation Rate).	CIO/CDTO, Chief Architect, PMO, Product Owners.	Прозрачная картина текущего состояния и динамики изменения техдолга.
Ретроспектива и улучшение процесса	▪ Ежеквартальные ретроспективы с командами и архитекторами по вопросам эффективности устранения техдолга. ▪ Корректировка подходов, стандартов и инструментов работы с техдолгом.	Архитектурные гильдии, Product Owners, CIO/CDTO.	Постоянное улучшение процесса, повышение зрелости команд и снижение накопления нового техдолга.

Построение карты технологического долга

Шаг процесса	Описание	Участники	Результат
Идентификация и регистрация техдолга	▪ Сбор данных о потенциальном техдолге (от команд разработки, поддержки, архитектуры). ▪ Регистрация в едином реестре техдолга (например, Jira, Confluence).	Архитекторы, команды разработки, Product Owners.	Заполненный и актуальный реестр техдолга с базовыми метриками (критичность, влияние, причина).
Классификация и приоритизация	▪ Определение категорий техдолга (архитектурный, кодовый, инфраструктурный, процессный). ▪ Приоритизация техдолга по критериям: ▪ Влияние на бизнес (время релизов, стабильность, затраты).	Chief Architect, Архитектурные гильдии, Product Owners.	Актуализированный реестр с расставленными приоритетами (high, medium, low).
Планирование работ с техдолгом			…й план задач по …олга, включённый в …и backlog.
Реализация и контроль устранения			…жение общего уровня …жение целевых KPI.
Мониторинг и анализ прогресса	▪ Регулярная отчетность (например, ежемесячно, ежеквартально) по уровню техдолга, KPI и достигнутым результатам. ▪ Оценка эффективности процесса управления техдолгом (метрики: Legacy Ratio, Tech Debt Remediation Rate).	CIO/CDTO, Chief Architect, PMO, Product Owners.	Прозрачная картина текущего состояния и динамики изменения техдолга.
Ретроспектива и улучшение процесса	▪ Ежеквартальные ретроспективы с командами и архитекторами по вопросам эффективности устранения техдолга. ▪ Корректировка подходов, стандартов и инструментов работы с техдолгом.	Архитектурные гильдии, Product Owners, CIO/CDTO.	Постоянное улучшение процесса, повышение зрелости команд и снижение накопления нового техдолга.

Система	Устаревание	Сложность поддержки	Влияние на бизнес	Общий риск
CRM	4/5	5/5	5/5	🔴 Высокий
DWH	3/5	3/5	4/5	🟠 Средний
Сайт	2/5	2/5	2/5	🟢 Низкий

Институционали-зировать Architecture Decision Records

Цель:

Зафиксировать и стандартизировать ключевые архитектурные решения, сделать их прозрачными и переиспользуемыми

Задачи:

1. Внедрить шаблон ADR.
2. Обязать команды описывать решения.
3. Сделать базу знаний по архитектуре доступной.

План действий:

1. Внедрить шаблон ADR (контекст, решение, альтернативы, статус).
2. Встроить ADR в процесс согласования и ревью.
3. Использовать Git, Confluence или Log4Brains для хранения.
4. Провести обучение команд.

Результат:

1. Осознанные решения → меньше хаоса.
2. Упрощение масштабирования и онбординга новых людей.
3. Возможность ретроспективно анализировать качество решений.

Институционализировать Architecture Decision Records

№ ADR	Дата принятия	Название решения	Контекст	Решение	Альтернативы	Статус	Владелец
001	12.01.2025	Стандартизация API на REST + JSON	Высокие издержки поддержки интеграций	RESTful API, JSON	SOAP, XML	Принято	ФИО
002	05.02.2025	Выбор Kubernetes как платформы контейнеризации	Потребность в масштабируемой инфраструктуре	Kubernetes (EKS/Azure AKS)	Docker Swarm, OpenShift	Принято	ФИО
003	17.03.2025	Переход на Apache Kafka как событийную платформу	Рост числа интеграций, низкая гибкость RabbitMQ	Apache Kafka (Confluent Platform)	RabbitMQ, ActiveMQ, IBM MQ	Принято	ФИО
004	28.03.2025	Выбор OAuth 2.0 + JWT как стандарта авторизации	Единая модель безопасности API	OAuth 2.0 + JWT	API Keys, Basic Auth	Принято	ФИО
005	15.04.2025	Переход на ELMA365 Low-code платформу	Сокращение времени разработки бизнес-приложений	Low-code платформа ELMA365	Custom Java/.NET решения	Принято	ФИО
006	10.05.2025	Выбор Terraform для Infrastructure-as-Code	Усложнение управления инфраструктурой	Terraform для IaC	Ansible, CloudFormation	Принято	ФИО
007	24.05.2025	Внедрение Data Lakehouse на основе Databricks	Необходимость эффективной работы с Big Data	Databricks Lakehouse	Hadoop, Snowflake, Amazon Redshift	Анализ	ФИО
008	01.06.2025	Введение Grafana и Prometheus как Observability	Недостаточный уровень мониторинга систем	Grafana + Prometheus	Zabbix, Elastic Stack	Принято	ФИО
009	07.06.2025	Использование ELK-стека для централизованных логов	Разрозненное хранение и анализ логов	ELK (Elastic Stack)	Splunk, Graylog	Принято	ФИО
010	15.06.2025	Архитектура микросервисов на основе DDD	Неуправляемость монолита и сложности поддержки	Domain Driven Design + микросервисы	Монолит, SOA	Принято	ФИО

Обеспечить участие архитекторов в продуктовых командах

Цель:

Сделать архитектуру частью delivery-процесса, а не внешним контролёром

Задачи:

1. Укрепить взаимодействие архитекторов и продакт-команд.
2. Минимизировать "архитектуру в вакууме".
3. Сократить количество архитектурных противоречий.

План действий:

1. Назначить архитекторов на кластеры / продукты.
2. Обеспечить их участие в PI-планированиях, backlog grooming, code review.
3. Создать каналы обратной связи: архитектурные гильдии, демо, архитектурные «стендапы».

Результат:

1. Архитектура — не тормоз, а поддержка.
2. Команды лучше понимают ограничения и стандарты.
3. Выравнивание скорости и устойчивости.

Связать бюджетирование и техдолг

Цель:

Превратить устранение техдолга в управляемое инвестиционное решение

Задачи:

1. Оценить бизнес-влияние техдолга.
2. Сделать его элементом инвестиционного планирования.
3. Защитить бюджет под архитектурные инициативы.

План действий:

1. Совместно с командами PMO/финансов описать техдолг в формате бизнес-кейса (влияние на скорость, ошибки, затраты).
2. Включить бюджетную статью "архитектурный редизайн" или "переход на целевую платформу".
3. Использовать формат "Refactoring Investment" с оценкой TCO.

Результат:

1. Архитектурные инициативы перестают быть «тратами» и становятся инвестициями.
2. Техдолг смотрят на равных с фичами и инициативами.

Внедрить метрики устойчивого развития ИТ

Цель:

Сделать устойчивость и зрелость ИТ измеримой

Задачи:

1. Ввести метрики архитектурной зрелости.
2. Связать их с OKR / KPI команд.
3. Использовать метрики в управлении изменениями и бюджетами.

План действий:

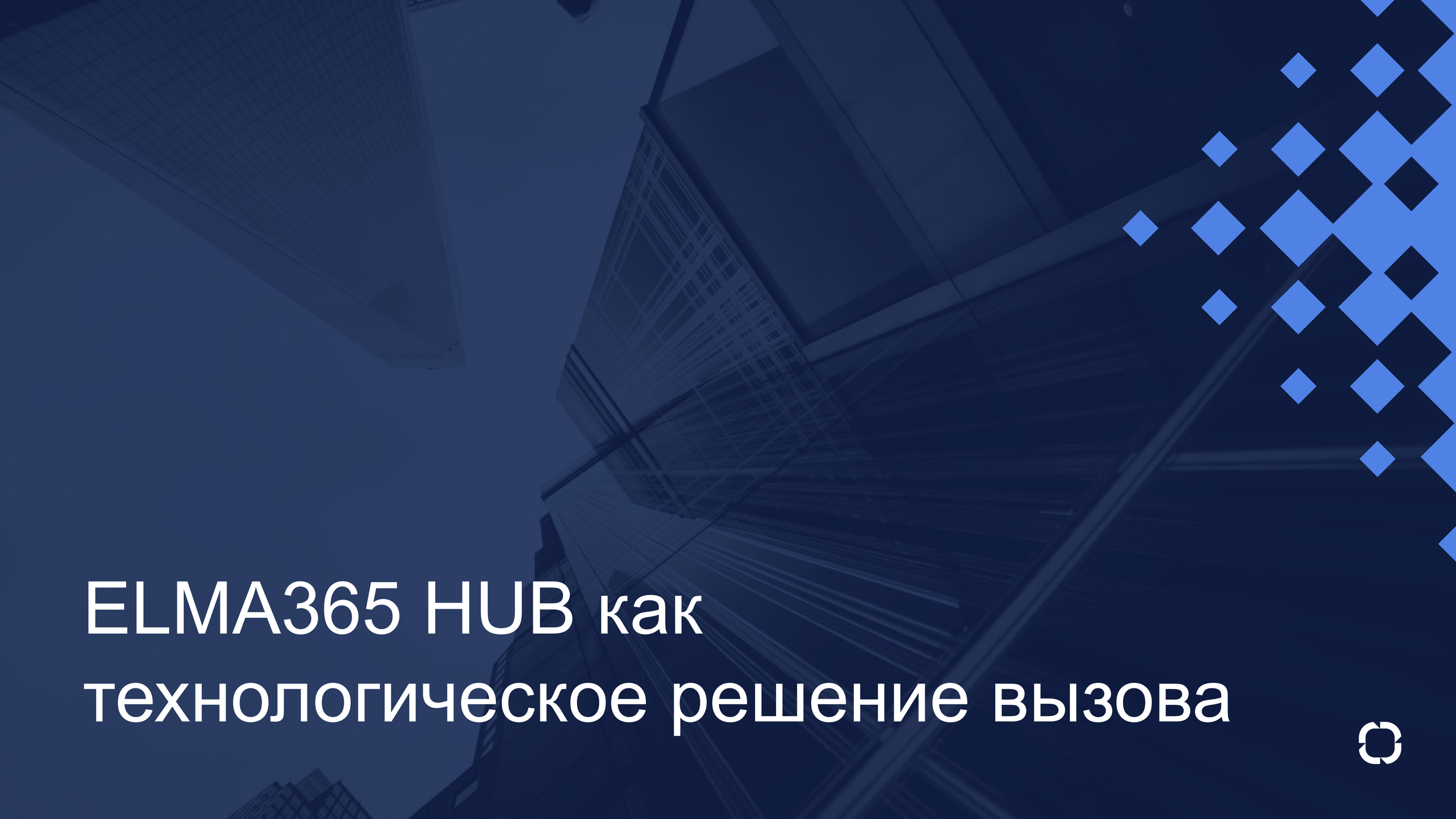
1. Внедрить: Change Lead Time, Legacy %, Tech Debt Remediation Rate, ADR Coverage.
2. Настроить визуализацию в BI-инструментах (например, Power BI, Grafana).
3. Включить метрики в отчётность на уровне CIO и Digital Office.

Результат:

1. Прозрачная картина зрелости ИТ-ландшафта.
2. Метрики как аргументы при споре скорости и качества.
3. Улучшение культуры принятия решений.

Внедрить метрики устойчивого развития ИТ

Архитектурные метрики			
Legacy Ratio	Доля устаревших (legacy) компонентов	(Кол-во Legacy-компонентов / общее число компонентов) × 100%	<20%
Architecture Compliance Ratio	Соответствие решений целевой архитектуре	(Количество решений, соответствующих архитектуре / общее количество решений) × 100%	>90%
ADR Coverage	Доля ключевых решений, оформленных в виде ADR	(Количество решений с ADR / общее количество решений) × 100%	>90%
API Reuse Rate	Переиспользование API и микросервисов	(Количество повторно используемых API / общее количество API) × 100%	↑ ежегодно на 10%
Метрики техдолга			
Tech Debt Remediation Rate	Доля устранённых задач техдолга	(Устраненные задачи техдолга / общее количество задач техдолга) × 100%	↑ >75%
Cost of Tech Debt	Оценка затрат на поддержку и устранение последствий техдолга	Σ (человеко-часы устранения последствий техдолга × средняя ставка часа)	↓ ежегодно на 10%
Tech Debt Registration Time	Среднее время от появления до регистрации долга в реестре	Среднее время (дней)	↓ до 7 дней
Метрики скорости и качества разработки			
Change Lead Time	Среднее время от идеи до реализации	Среднее время (дни)	↓ ежегодно на 15–20%
Deployment Frequency	Частота релизов на прод	Количество релизов / период	↑ не менее 1 раза в неделю
Change Failure Rate	Доля релизов с ошибками или откатами	(Количество релизов с ошибками / общее число релизов) × 100%	<5%
Mean Time To Recovery (MTTR)	Среднее время восстановления после сбоев	Среднее время восстановления (часов)	↓ <2 часов
Метрики инфраструктуры и операционной эффективности			
Availability Rate (SLA)	Доступность критических сервисов	(Общее время работы без сбоев / общее время эксплуатации) × 100%	>99,9%
Infrastructure Automation Level	Доля инфраструктуры, управляемой через IaC	(Инфраструктура в IaC / общая инфраструктура) × 100%	↑ до 80%
Incidents per Service	Среднее количество инцидентов на один сервис	(Количество инцидентов / общее число сервисов)	↓ ежегодно на 20%



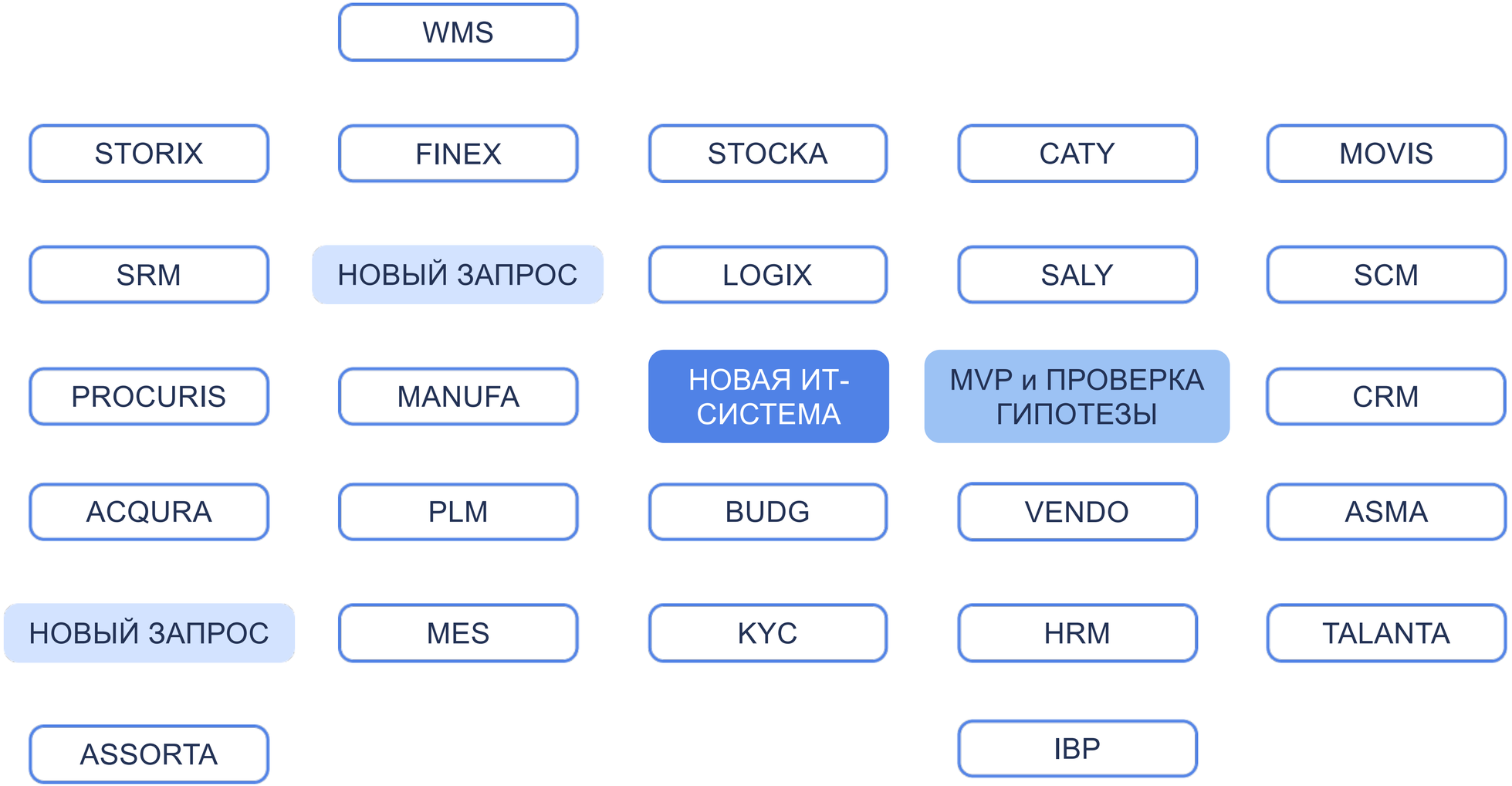
ELMA365 HUB как технологическое решение вызова

Текущая ситуация

STORIX	WMS	STOCKA	CATY	MOVIS
SRM	FINEX	LOGIX	SALY	SCM
PROCURIS	MANUFA	НОВАЯ ИТ-СИСТЕМА	VENDO	CRM
ACQURA	PLM	BUDG	HRM	ASMA
ASSORTA	MES	KYC	IBP	TALANTA

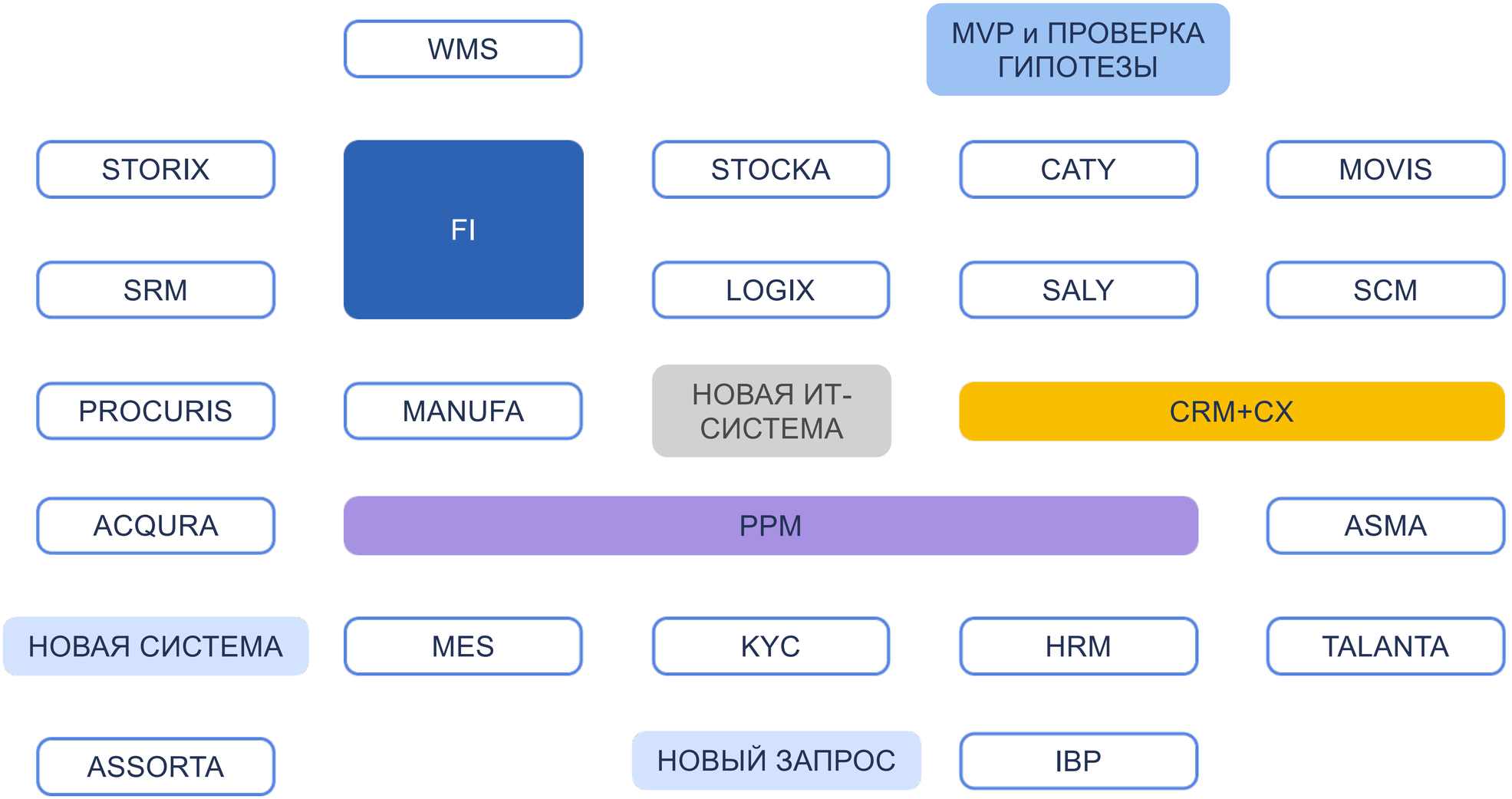
Системный ландшафт не является статичным. У бизнеса появляются новые запросы, MVP и гипотезы.

Среднесрочная перспектива



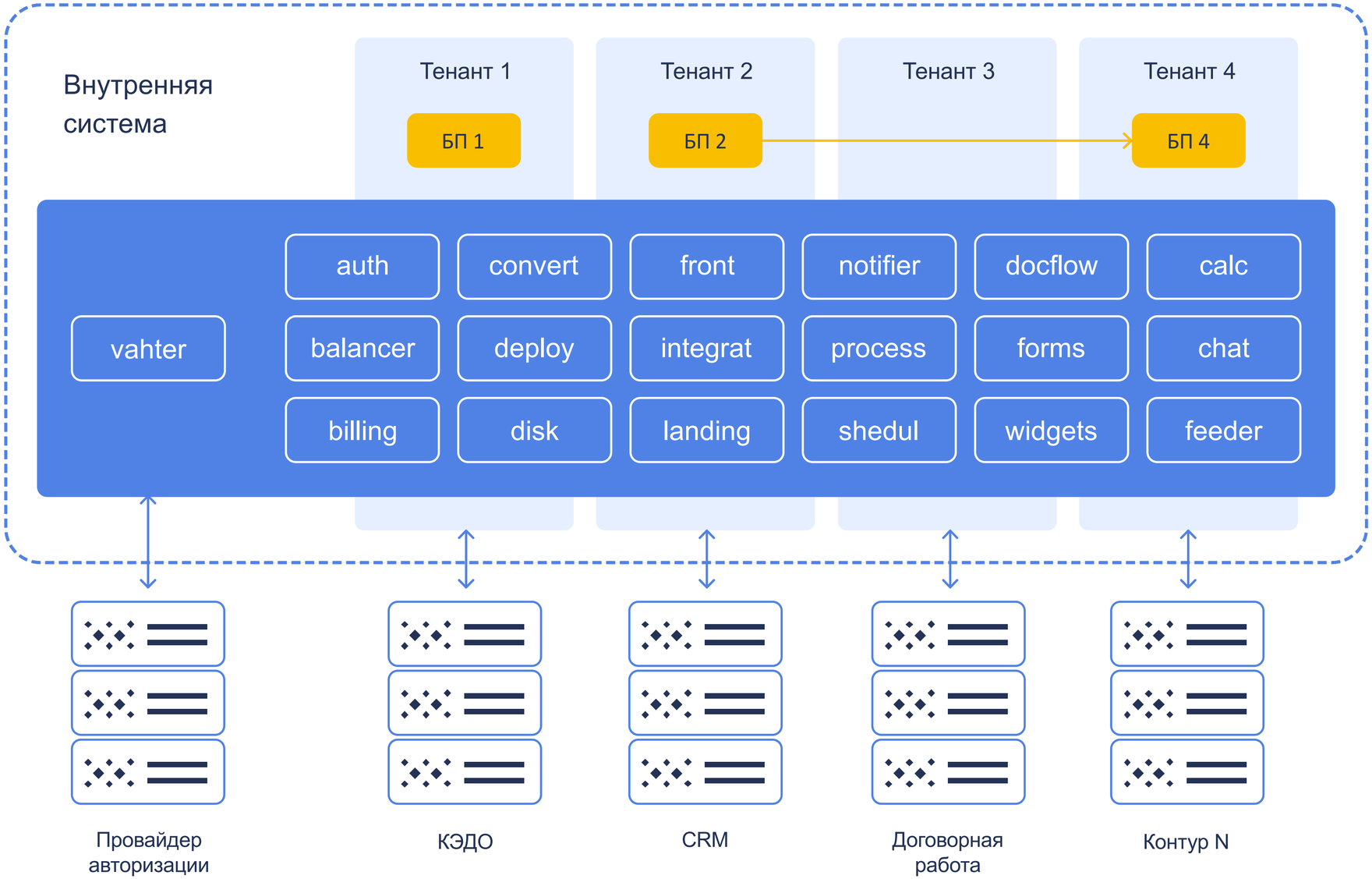
Системный ландшафт не является статичным. У бизнеса появляются новые запросы, MVP и гипотезы.

Долгосрочная перспектива



Системный ландшафт не является статичным. У бизнеса появляются новые запросы, MVP и гипотезы.

ELMA365 HUB



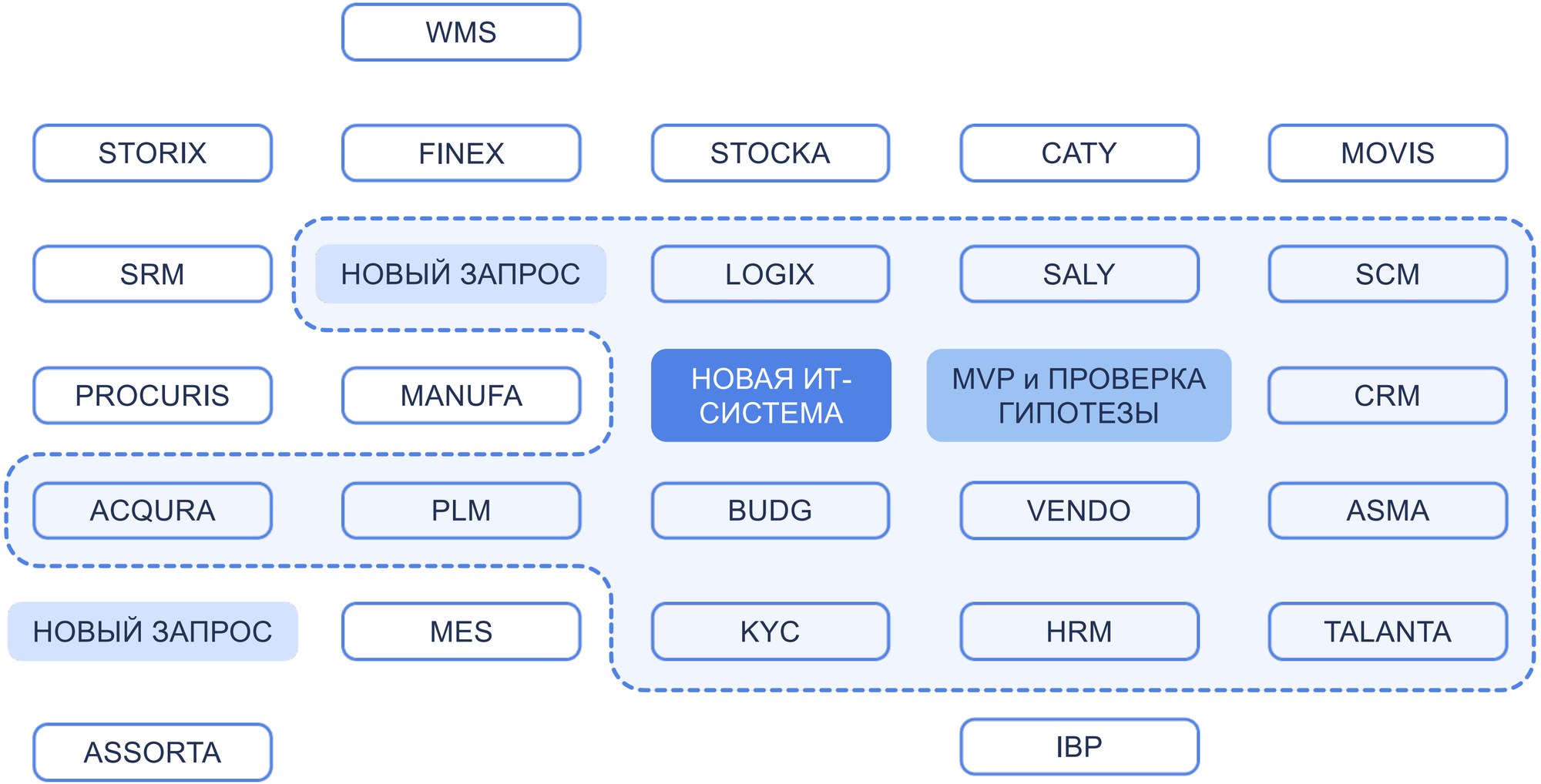
ELMA365 HUB – это уникальный вид поставки ПП ELMA365.

Компания получает возможность в рамках одного вычислительного кластера создать неограниченное количество сред разработки для разных проектов (тенантов).

Тенанты изолированы друг от друга логически и на уровне данных. Это обеспечивает асинхронный CI/CD цикл для разных тенантов (разных проектов).

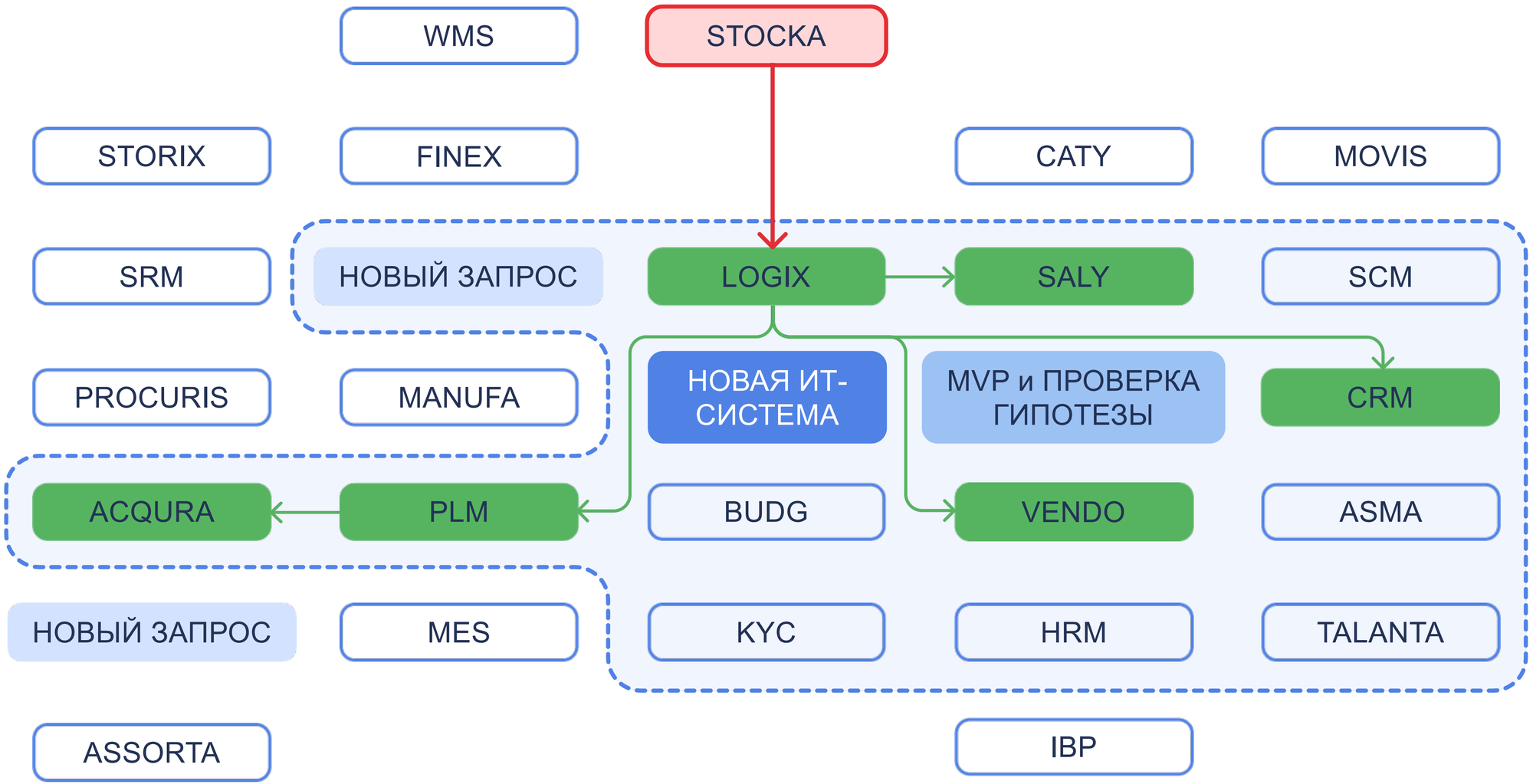
Изоляция данных создает дополнительный уровень независимости, при этом тенанты могут быть связаны сквозными бизнес-процессами.

ELMA365 HUB



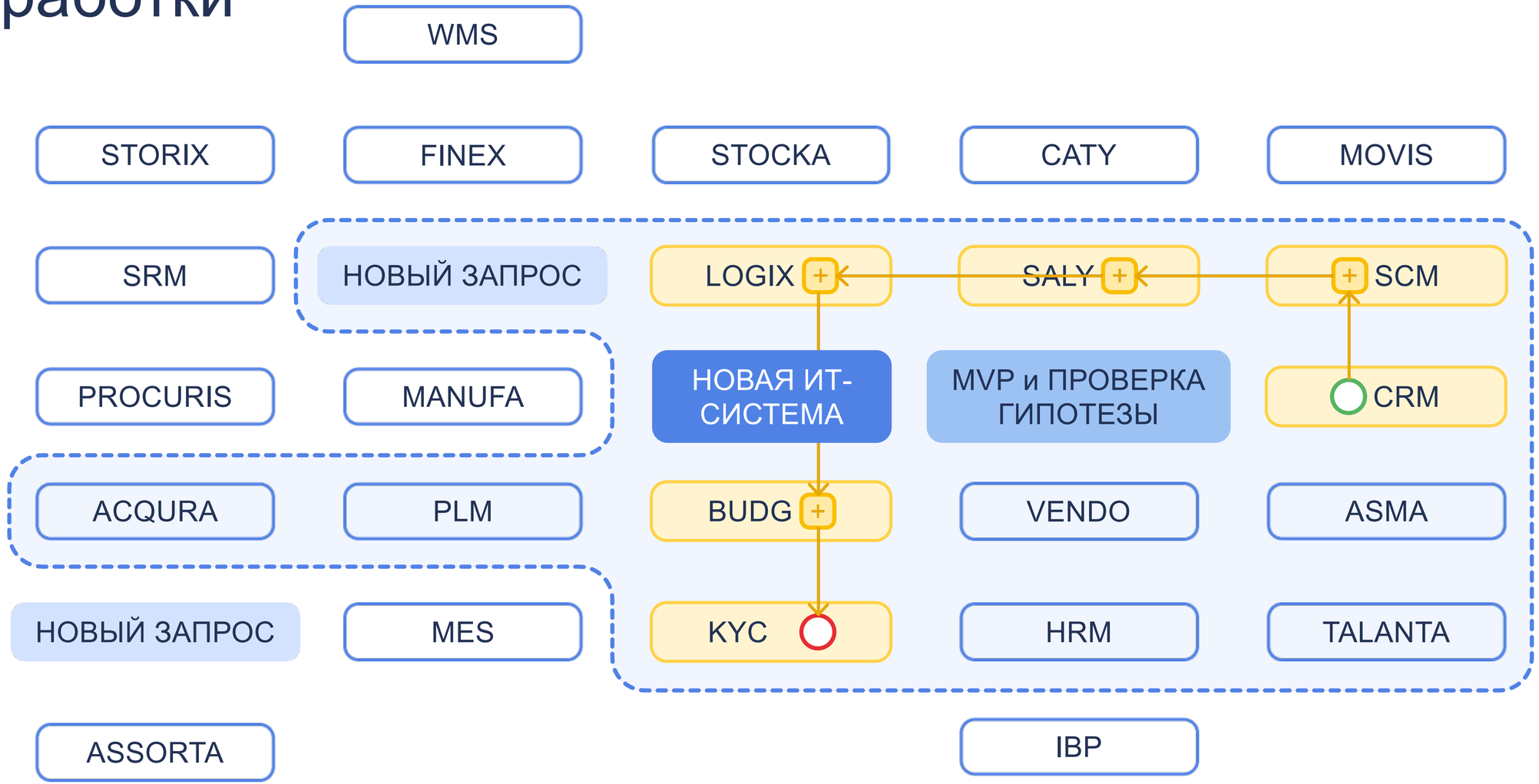
ELMA365 HUB может быть использован как для замещения устаревших ИТ-систем (снижает уровень Legacy), так и создания новых систем по запросам бизнеса.

Отсутствие интеграций снижает Time-to-Market систем



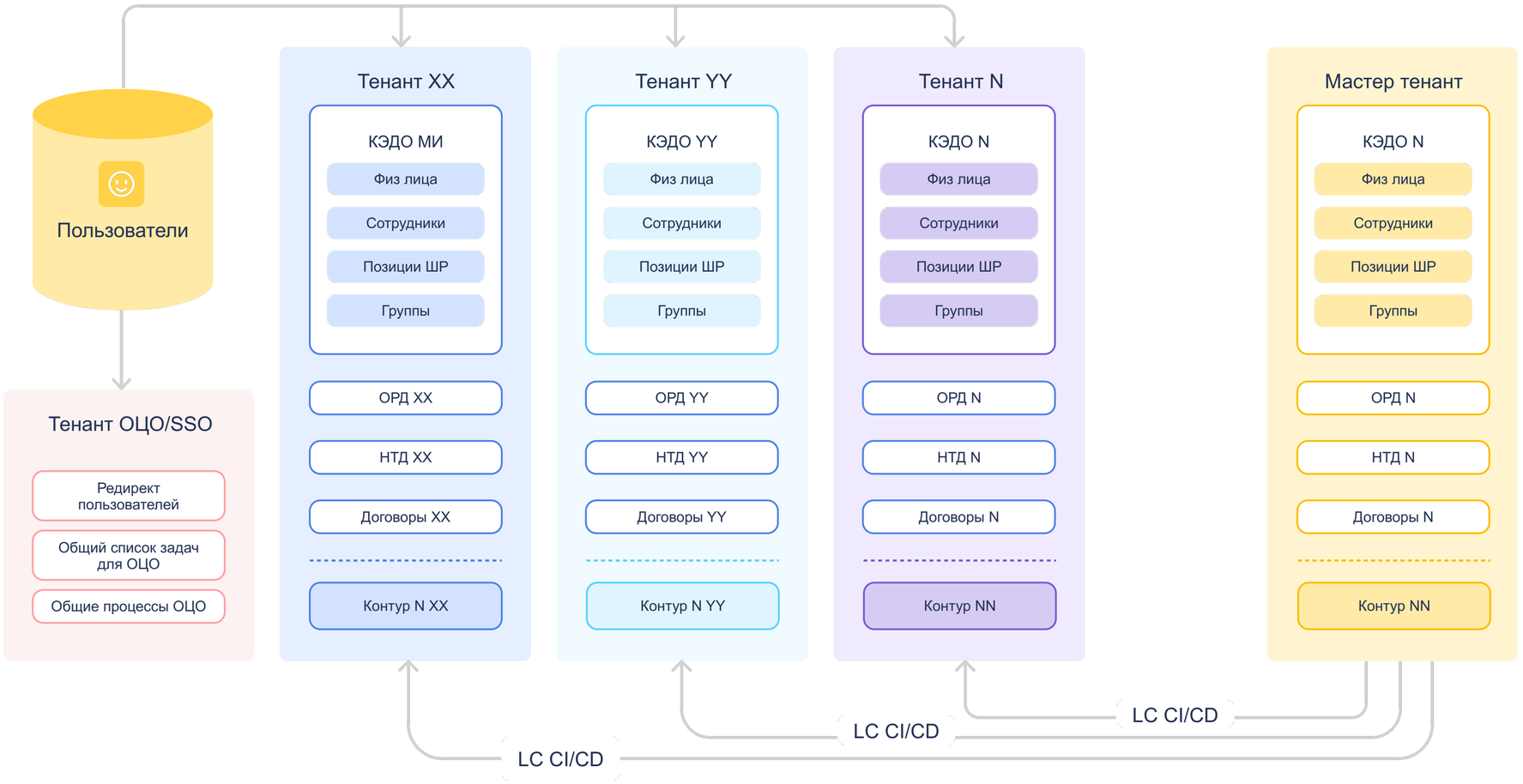
Работа с данным в ELMA365 HUB обеспечивает отсутствие интеграций при появлении новой системы.
Тенанты могут «делаться информацией» с соседями по запросу.

Сквозные процессы повышают эффективность разработки



Сквозные бизнес-процессы внутри ELMA365 обеспечивают связность систем на уровне бизнес-логики. Так процесс может быть запущен в одной системе (тенанте), а продолжиться в другом.

ELMA365 HUB в холдинге



Применение ELMA365 HUB в холдинговых компаниях закрывает несколько ключевых сценариев использования:

1. Создание внутреннего реестра готовых типовых решений, которые могут использоваться и адаптироваться в тенантах дочерних обществ;
2. Обмен данными между тенантами дочерних обществ и головной компанией;
3. Выстраивание сквозных бизнес-процессов между дочерними обществами и головной компанией.

Эффекты использования ELMA365 HUB

- Единое архитектурное решение для бизнес-автоматизации
- Решение новых задач автоматизации на единой платформе с минимальным Time-to-Market
- Инструмент снижения техдолга – перенос Legacy решений без ограничений за счет Low- и Hard- code
- Выстраивание и поддержка процессов асинхронного CI/CD для продуктовых команд

БИЗНЕС В ЦИФРЕ

Экспертный
телеграм-канал
Андрея Чепакина

Цифровая трансформация — один из самых обсуждаемых трендов последних лет. Каждый трактует её по-своему: для одних это внедрение ИИ, для других – автоматизация или переход в облако.

Однако за этими модными терминами часто теряется главное: **цифровая трансформация — это про радикальное переосмысление управленческих стратегий, операционных моделей и методов взаимодействия с заинтересованными сторонами.**

Мой телеграм-канал — для тех, кто хочет видеть в ЦТ не просто технологические «игрушки», а реальный инструмент для роста эффективности, масштабирования и управления изменениями.

Меня зовут Чепакин Андрей, и моя экспертиза — это управление бизнес-процессами. Я убеждён, что ЦТ невозможна без глубокого понимания того, как устроена и функционирует компания. Только через эту призму можно увидеть реальные точки изменений, а не просто следовать за трендом.

Если ваш фокус — не просто технологии, а управляемые изменения, подписывайтесь и присоединяйтесь к обсуждению. Мы говорим о главном.

Вопросы