EnvSpec Naming

Открытый стандарт семантического именования ИТ-ресурсов.

1. Назначение стандарта

Стандарт устанавливает единую систему координат для ИТ-ресурсов организации: иерархическую модель «окружение → контур → стенд → слот → узел», правила построения сетевых имён (DNS/FQDN), криптографических идентификаторов (SPIFFE), идентификаторов плоских сред (Kubernetes Namespaces, облачные проекты, объекты СУБД) и обязательные теги инвентаризации.

Стандарт решает следующие задачи:

- определяет единую семантику понятий для бизнеса, разработки, эксплуатации и информационной безопасности
- вводит машиночитаемость имён и тегов для инвентаризации (CMDB), учёта затрат (FinOps) и политик доступа ИБ
- запрещает кодирования изменчивых свойств (версии, SLA, площадки) в именах ресурсов
- определяет положения платформенных сервисов, рабочих мест сотрудников и внешних систем в модели доверия

Стандарт не определяет процессы согласования доступов, состав политик безопасности и порядок их применения.

2. Термины и определения

> **КЛЮЧЕВЫЕ СЛОВА**
>
> Ключевые слова **ДОЛЖЕН** (MUST), **НЕ ДОЛЖЕН** (MUST NOT), **СЛЕДУЕТ** (SHOULD), **НЕ СЛЕДУЕТ** (SHOULD NOT) и **МОЖЕТ** (MAY) трактуются в соответствии с RFC 2119.

- **Цифровой актив (Digital Asset)** — информационная система, сервис или платформенный компонент, учтённый в корпоративном реестре под уникальным мнемокодом.
- **Окружение (Environment)** — верхнеуровневый режим доверия, определяемый двумя признаками: *кто потребители* и *какие данные обрабатываются*. Характер данных однозначно задаётся окружением.
- **Контур (Perimeter)** — именованная совокупность стендов одного окружения, объединённых для сквозного взаимодействия под конкретную бизнес- или технологическую задачу (продуктовое направление, комплаенс-периметр, пилот, платформенные сервисы). Контур — единица управления.
- **Стенд (Instance)** — развёрнутый экземпляр цифрового актива внутри контура, включающий исполняемую часть и собственные ресурсы.
- **Слот (Slot)** — функциональная часть стенда. Логический слой или релизная группа (`api`, `worker`, `db`, `ui`, `v2`). Используется для разграничения ролей узлов и параллельного развёртывания версий (blue-green, канареечные релизы).
- **Узел (Node)** — единица вычислительной инфраструктуры, на которой работает слот стенда: виртуальная машина, физический сервер, worker-узел кластера.
- **Площадка (Site)** — физическое место размещения узла: ЦОД, облачный регион, зона доступности. В иерархию не входит.
- **Сегмент (Segment)** — техническое средство изоляции ресурсов на сетевом (VLAN, подсеть, файрвол), платформенном (Network Namespace, Service Mesh) или логическом (Kubernetes Namespace, тег доступа) уровне. Сегмент реализует границы окружения, контура, стенда, слота или узла, но не определяет их логику. В иерархию не входит. Совместное размещение ресурсов в одном сегменте не делает их одним стендом, слотом или контуром.
- **Субъект доступа (Subject)** — сотрудник или автоматизированный процесс, обращающийся к цифровым активам. Идентифицируется учётной записью корпоративного каталога. В иерархию не входит.
- **Рабочее место (Workplace)** — устройство или сессия, с которых сотрудник обращается к цифровым активам: АРМ, ноутбук, мобильное устройство, VDI-сессия.
- **Шлюз доступа (Access Gateway)** — стенд, через который субъекты получают доступ к цифровым активам окружения: опубликованный пользовательский интерфейс, API-шлюз, bastion/PAM, VDI-брокер, VPN, ZTNA-прокси.

- **Внешняя система (External System)** — цифровой актив вне зоны контроля организации: SaaS, система контрагента, государственная система.

Если компоненты одной информационной системы требуют размещения в разных режимах доверия (например, DMZ и внутренний периметр), такие компоненты ДОЛЖНЫ регистрироваться в корпоративном реестре как отдельные цифровые активы со своими уникальными мнемокодами (например, `{system}-ui` и `{system}-api`).

3. Иерархическая модель

Стандарт базируется на строгой иерархии пяти уровней, следующей принципу «от частного к общему»:

```
Окружение (prod)                          ← режим доверия и характер
данных
└── Контур (dmz)                          ← логический периметр под
задачу
    └── Стенд (api-management)            ← экземпляр цифрового
актива
        └── Слот (websocket-ingress)      ← роль или релизный слот
внутри стенда
            └── Узел (ingress-01)         ← вычислительная единица
```

Иерархия строгая: узел принадлежит ровно одному слоту, слот — ровно одному стенду, стенд — ровно одному контуру, контур — ровно одному окружению. Площадка, субъект доступа и рабочее место уровнями иерархии не являются.

3.1. Окружения

Стандарт фиксирует шесть идентификаторов окружений: `dev`, `test`, `stage`, `prod`, `infrastructure`, `workplace`. Список окружений НЕ ДОЛЖЕН расширяться: потребность в особой среде (пилот, песочница, нагрузочное тестирование, архив, приёмка) ДОЛЖНА выражаться контуром внутри одного из окружений.

Окружения `dev`, `test`, `stage`, `prod` образуют линейную шкалу возрастания режима доверия:

```
dev < test < stage < prod
```

Окружения `infrastructure` (3.2) и `workplace` (3.3) в линейную шкалу не входят.

Критерий принадлежности к окружению. Принадлежность определяется потребителями и характером данных, а НЕ мощностью оборудования, площадкой, стоимостью или стадией эксплуатации информационной системы:

Идентификатор	Основные потребители	Характер данных
`dev`	Разработчики	Синтетические данные
`test`	Команды тестирования (QA), смежные команды интеграции	Синтетические или обезличенные данные
`stage`	Релиз-инженеры, команды приёмки на обезличенных данных	Конфигурация — зеркало `prod`. Только синтетические или обезличенные данные
`prod`	Конечные пользователи, автоматизированные бизнес-процессы, уполномоченный персонал с правами доступа к реальным данным (приёмка, миграции, сопровождение)	Реальные (продуктовые) данные
`infrastructure`	Инженерный персонал (DevOps, сопровождение, ИБ), автоматизированные процессы доставки, наблюдаемости и управления доступом	Технологические данные всех окружений: секреты, артефакты, журналы, метрики, резервные копии, учётные записи. Режим защиты — не ниже `prod`
`workplace`	Сотрудники всех ролей	Собственных данных ландшафта не хранит. Данные окружений доступны только через шлюзы доступа (3.5)

- Характер данных — функция окружения. Контур, обрабатывающий реальные данные, принадлежит `prod` по определению, как бы он ни назывался и для кого бы ни создавался: это в равной мере относится к приёмке (UAT), опытной эксплуатации и техническим миграциям на реальных данных. В частности, пилот, читающий реальные данные без права их изменения, ДОЛЖЕН выполняться в отдельном контуре `prod`, а не в `stage` «с исключением».
- Реальные данные НЕ ДОЛЖНЫ попадать в `dev`, `test` и `stage` без предварительного обезличивания.
- Окружение не задаёт критичность, SLA, мощность и состав резервирования. Контур `prod` с низкой критичностью и одной репликой допустим. Недопустимо ослабление мер защиты реальных данных, единых для всех контуров `prod`.

3.2. Окружение `infrastructure`

Окружение платформенных цифровых активов, обслуживающих все окружения ландшафта: CI/CD, реестры артефактов, хранилища секретов, корпоративные каталоги (IdP/AD), мониторинг и сбор журналов, резервное копирование, управление конфигурацией, управление рабочими местами.

- Режим доверия `infrastructure` — не ниже `prod`: компрометация платформенного стенда равнозначна компрометации `prod`. Меры защиты стендов `infrastructure` НЕ ДОЛЖНЫ быть слабее мер защиты `prod`.
- Стенд `infrastructure` НЕ ДОЛЖЕН служить каналом передачи данных между окружениями линейной шкалы (например, `prod` → `test` через общий брокер, реестр или хранилище журналов). Данные разных окружений внутри стенда `infrastructure` ДОЛЖНЫ быть разделены средствами самого актива (арендаторы, индексы, бакеты, проекты) с сохранением признака окружения-источника.
- Стенд `infrastructure` МОЖЕТ размещать системы сопровождения и системы информационной безопасности (SIEM, EDR, DLP, PAM). Сбор, обработка и хранение событий аудита ИБ ДОЛЖНЫ выполняться в `prod` цифровыми активами (в том числе платформенными) окружения.
- Экземпляры платформенных систем, создаваемые для разработки и тестирования самих этих систем, размещаются как обычные стенды в `dev`, `test`, `stage` (`vault.secrets.test`), а не в `infrastructure`.
- Для платформенных систем автоматизации и хранения данных (резервное копирование, сбор журналов, объектные хранилища) допускается разделение

архитектуры: компоненты управления (Control Plane) размещаются в `infrastructure`, а компоненты хранения продуктовых данных (Data Plane / Репозитории) размещаются в контурах соответствующих целевых окружений (например, `prod`), являясь частью их инфраструктуры.

3.3. Окружение `workplace`

Окружение рабочих мест сотрудников: АРМ, ноутбуки, мобильные устройства, VDI-сессии, а также инфраструктура, непосредственно их обслуживающая (VDI-брокеры, терминальные фермы, VPN-концентраторы).

- Рабочее место — недоверенное устройство. Оно НЕ ДОЛЖНО получать доверие по факту сетевого расположения и НЕ ДОЛЖНО быть местом хранения данных окружений линейной шкалы и `infrastructure`.
- Роль сотрудника (разработка, тестирование, DevOps, сопровождение, ИБ, внутренний пользователь информационных систем) НЕ ДОЛЖНА кодироваться в окружении, контуре или имени рабочего места. Права субъекта определяются его учётной записью и ролями в корпоративном каталоге (стенд `infrastructure`), а не рабочим местом.
- Доступ с рабочего места к цифровым активам любого окружения, включая `infrastructure`, осуществляется только через шлюз доступа этого окружения (3.5).
- `{perimeter}` рабочего места — группа рабочих мест по способу управления и подключения (например `office`, `remote`, `vdi`, `kiosk`); `{system}` — код класса рабочих мест из реестра цифровых активов (например `arm`, `laptop`, `vdi`). Шаблон FQDN (4.3) МОЖЕТ применяться к рабочим местам, разметка тегами (раздел 6) обязательна.
- Исключением из правила недоверенности рабочих мест являются выделенные АРМ технологического управления (сетевые инженеры, ИБ-администраторы) и Air-Gap терминалы. Такие устройства ДОЛЖНЫ выделяться в отдельные контуры окружения `workplace` (например, `net-ops`, `sec-ops`, `airgap`). Сетевой доступ к критической инфраструктуре (сетевое оборудование, ядра систем) ДОЛЖЕН ограничиваться на уровне межсетевых экранов по признаку принадлежности к этим технологическим контурам.
- Для АРМ, находящихся в физически изолированных сегментах (Air-Gap) и не имеющих доступа к корпоративной службе каталогов и DNS, канонический шаблон FQDN применяется локально с заменой корпоративного суффикса

`{domain}` на локальный (например, `.local`). При этом разметка обязательными тегами инвентаризации в корпоративной CMDB является строго обязательной, а тег `envspec.io/site` ДОЛЖЕН содержать точную локацию физического размещения (здание, комната, гермозона).

3.4. Зарезервированные контуры `platform` и `external`

В каждом окружении зарезервированы два идентификатора контура. Они НЕ ДОЛЖНЫ использоваться для иных целей.

- **`platform`** — контур платформенных цифровых активов окружения, обслуживающих несколько контуров этого же окружения: разделяемые кластеры СУБД, брокеры сообщений, ESB и API-шлюзы, кластеры оркестрации, сервисы аутентификации приложений, шлюзы доступа, системы ИБ и так далее. Цифровой актив, обслуживающий несколько контуров, размещается один раз в `platform`, стенды других контуров обращаются к нему как клиенты. Разграничение потребителей выполняется средствами самого актива (схемы, топики, пространства имён), а не разделением стенда на контуры.
- **`external`** — контур представления внешних систем, сопряжённых с данным окружением. Внешняя система получает идентификатор `{system}.external.{env}` в том окружении, с данными которого она обменивается: боевой API контрагента — в `external.prod`, его песочница — в `external.test`. Идентификатор служит для локальных DNS-алиасов, правил шлюзов, тегов и деклараций; он не является собственным именем внешней системы. Стенды организации НЕ ДОЛЖНЫ размещаться в контуре `external`.

3.5. Шлюзы доступа

- Субъекты доступа с рабочих мест и внешние системы получают доступ к цифровым активам окружения только через шлюзы доступа. Прямая сетевая связность рабочих мест и внешних систем с узлами стендов, минуя шлюз, ЗАПРЕЩЕНА.
- Шлюз доступа — стенд того окружения, в которое он открывает доступ, и размещается в его контуре `platform` либо в контуре обслуживаемой системы: `bastion.platform.prod`, `ingress.platform.prod`, `apigw.platform.prod`, `sso.platform.infrastructure`. VPN-концентраторы и VDI-брокеры

расширяют окружение `workplace` на удалённые устройства и принадлежат `platform.workplace`.

- Шлюз ДОЛЖЕН аутентифицировать субъекта по учётной записи корпоративного каталога. Матрица «роль субъекта → окружение → цифровой актив» — предмет политики организации и стандартом не определяется, стандарт фиксирует точку входа (шлюз) и источник идентичности (каталог).

3.6. Правила доверия между окружениями

1. Стенды разных окружений линейной шкалы (`dev`, `test`, `stage`, `prod`) НЕ ДОЛЖНЫ взаимодействовать напрямую.
2. `infrastructure` — единственное окружение, стендам которого разрешено взаимодействие со стендами любого окружения. Допускаются два класса взаимодействий: **управляющее** (`infrastructure` → окружение: доставка артефактов и конфигурации, выдача секретов, управление учётными записями и рабочими местами) и **телеметрическое** (окружение → `infrastructure`: журналы, метрики, трассировки). Ограничение транзита — по 3.2.
3. `workplace` → любое окружение, включая `infrastructure`: только через шлюз доступа целевого окружения (3.5).
4. `external.{env}` ↔ стенды `{env}`: только через шлюз доступа (ingress/egress) окружения `{env}`. Взаимодействие `external.{env}` со стендами других окружений ЗАПРЕЩЕНО: стенд `prod` НЕ ДОЛЖЕН обращаться к `*.external.test`, и наоборот.
5. Порядок оформления и согласования исключений из правил 1–4 — вне области стандарта.

3.7. Ортогональность физического размещения

Площадка (ЦОД, облачный регион, зона доступности) — ортогональное измерение «где». Узел находится ровно на одной площадке; контур и стенд МОГУТ распределяться по нескольким. Площадка фиксируется в атрибутах инвентаризации (CMDB, теги). Код площадки НЕ ДОЛЖЕН включаться в сетевые имена и идентификаторы ресурсов. Перенос узла между площадками не должен приводить к изменению его имени.

4. Соглашение об именовании

4.1. Формат компонентов имени

- Каждый компонент имени ДОЛЖЕН соответствовать шаблону `^[a-z0-9]([a-z0-9-]*[a-z0-9])?$`: строчные латинские буквы, цифры, дефис внутри метки.
- Символ подчёркивания (`_`), заглавные буквы и спецсимволы в компонентах ЗАПРЕЩЕНЫ (исключение — разделитель плоских сред по разделу 5).
- Длина одного компонента НЕ ДОЛЖНА превышать 63 символов.
- `{perimeter}` и `{system}` ДОЛЖНЫ быть не длиннее 13 символов: это гарантирует вписывание всех плоских форм раздела 5 в лимит 63 символов. Для платформ с более жёсткими лимитами организация ДОЛЖНА установить меньший лимит.
- Декомпозированный мнемокод `{system}` (например, `{system}-ui`, `{system}-api` по разделу 2) МОЖЕТ достигать 16 символов при сохранении вписывания в лимит 63 символов плоских форм раздела 5.
- В `{perimeter}` и `{system}` НЕ СЛЕДУЕТ использовать дефис: плоские формы с дефисным разделителем (5.2, 5.3) при этом раскладываются на компоненты однозначно.
- `{env}` — строго один из идентификаторов: `dev`, `test`, `stage`, `prod`, `infrastructure`, `workplace`.
- Уникальность: `{perimeter}` уникален в пределах окружения, `{system}` — в пределах реестра цифровых активов, `{slot}` — в пределах стенда, `{node}` — в пределах слота.
- Компоненты следуют строго в иерархическом порядке **от частного к общему**; суффикс всегда определяет окружение.

4.2. Запрет изменчивых свойств

В именах ресурсов ЗАПРЕЩЕНО кодировать изменчивые эксплуатационные или организационные характеристики. Например, параметры SLA, классы критичности (~~tier1~~, ~~prod-critical~~), статусы жизненного цикла (~~legacy~~, ~~old~~), технические параметры оборудования, площадку, роль сотрудника.

Изменение критичности, топологии или площадки НЕ ДОЛЖНО приводить к переименованию ресурса или смене DNS-записи. Эти свойства ведутся в тегах и инвентаризации, а не в именах.

Мажорные версии и релизные группы (blue-green, канареечные релизы) ДОЛЖНЫ выражаться исключительно компонентом `{slot}` (`app-01.v2.antifraud.…`) и НЕ ДОЛЖНЫ склеиваться с кодом системы или контура (~~antifraud-v2~~).

4.3. Сетевые имена (DNS / FQDN)

Канонический шаблон FQDN инфраструктурного узла:

```
{node}.{slot}.{system}.{perimeter}.{env}.{domain}
```

- `{node}` — имя узла: функциональная роль и порядковый номер (`app-01`, `db-02`, `worker-11`).
- `{slot}` — функциональный компонент или релизный слой стенда (`api`, `worker`, `db`, `ui`, `v2`). Если стенд не требует внутреннего деления, компонент ДОЛЖЕН принимать значение `main`. Число сегментов имени фиксировано.
- `{system}` — мнемокод цифрового актива из корпоративного реестра.
- `{perimeter}` — идентификатор контура, включая зарезервированные `platform` и `external` (3.4).
- `{env}` — идентификатор окружения (4.1).
- `{domain}` — базовая корпоративная DNS-зона (`example.ru`).

Эталонные примеры:

FQDN	Чтение
`db-01.main.billing.payments.prod.example.ru`	узел СУБД №1 стенда `billing` в контуре `payments` окружения `prod`
`app-02.v2.antifraud.risk.test.example.ru`	второй узел релизного слота `v2` стенда `antifraud` в контуре `risk` окружения `test`

FQDN	Чтение
`runner-24.main.jenkins.cicd.infrastructure.example.ru`	раннер №24 стенда `jenkins` в контуре `cicd` окружения `infrastructure`
`pg-01.main.pgcluster.platform.prod.example.ru`	узел №1 разделяемого кластера PostgreSQL в платформенном контуре `prod`
`bastion-01.main.pam.platform.prod.example.ru`	шлюз привилегированного доступа в `prod`
`api.main.companyid.external.prod.example.ru`	локальный алиас точки интеграции боевого API контрагента, сопряжённого с `prod`
`mgmt-arm-01.main.arm.net-ops.workplace.example.ru`	выделенный APM №1 в слоте `main` для управления сетью (`mgmt-arm`) в контуре сетевых инженеров (`net-ops`) окружения рабочих мест
`term-02.main.terminal.airgap.workplace.local`	изолированный Air-Gap терминал №2 в контуре `airgap` локального ландшафта

Границы применения. Шаблон обязателен для инфраструктурных узлов (виртуальных машин, физических серверов, worker-узлов кластеров). Эфемерные нагрузки оркестрируемых платформ (поды, задачи) FQDN по шаблону не получают: их принадлежность иерархии несут пространство имён и метки платформы (раздел 5) и SPIFFE ID (4.4). Имя в контуре `external` — локальный алиас (CNAME или адрес шлюза), а не имя внешней системы. Географические DNS-зоны, если они нужны, оформляются техническими алиасами поверх канонического имени, а не его частью.

4.4. Криптографические имена (SPIFFE / Workload Identity)

При использовании взаимной аутентификации (mTLS) или токенов безопасности (JWT) иерархия имён ДОЛЖНА транслироваться в идентификаторы SPIFFE:

```
spiffe://{trust-
domain}/env/{env}/perimeter/{perimeter}/system/{system}/slot/{slot}
```

`{trust-domain}` — домен доверия организации (`example.ru`), остальные компоненты строго соответствуют компонентам DNS-имени.

Кодирование идентификатора в артефакты безопасности:

- **X.509-SVID:** идентификатор записывается в расширение Subject Alternative Name (SAN) типа URI, поле Subject Common Name (CN) дублирует канонический FQDN узла. Пример: CN `db-01.main.billing.payments.prod.example.ru`, SAN URI `spiffe://example.ru/env/prod/perimeter/payments/system/billing/slot/main`.
- **JWT-SVID:** идентификатор записывается в обязательное поле `sub`. Пример: `"sub": "spiffe://example.ru/env/prod/perimeter/payments/system/billing/slot/main"`.

Правила выдачи:

- Генератор идентичности НЕ ДОЛЖЕН выдавать SPIFFE ID только по данным от нагрузки. Компоненты `env`, `perimeter`, `system` ДОЛЖНЫ подтверждаться метаданными платформы размещения (обязательные теги, пространство имён и метки Kubernetes). Ресурс с тегом `env: test` не может получить идентификатор с `/env/prod/`.
- Внешним системам (`external`) идентификаторы домена доверия организации НЕ ДОЛЖНЫ выдаваться.
- Рабочим местам (`workplace`) идентификаторы по шаблону НЕ ДОЛЖНЫ выдаваться. Идентичность субъекта — учётная запись корпоративного каталога.

Специальные платформенные средства (Service Mesh) для применения правила не требуются. Идентификатор — строка, собираемая действующим генератором сертификатов (HashiCorp Vault, локальный PKI) и проверяемая регулярным выражением в коде приложения или конфигурации веб-сервера.

Идентификация рабочих мест (`workplace`) и субъектов доступа с помощью криптографических сертификатов X.509 (включая аппаратные токены и смарт-карты) подчиняется следующим правилам:

- Персональные сертификаты сотрудников идентифицируют субъекта доступа, привязываются к корпоративному каталогу (поля CN владельца, UserPrincipalName) и не используют иерархический шаблон инфраструктуры.
- Сертификаты устройств (машинные сертификаты) для АРМ и терминалов ДОЛЖНЫ содержать канонический FQDN рабочего места, построенный по правилам 4.3 (например, `CN=mgmt-arm-01.main.arm.net-ops.workplace.example.ru`).
- Шлюзы доступа ДОЛЖНЫ валидировать машинные сертификаты устройств и сопоставлять их контур (`{perimeter}`) с правами доступа субъекта. Доступ к управлению инфраструктурой с устройств, чьи сертификаты не принадлежат доверенным технологическим контурам (например, `net-ops` или `sec-ops`), ДОЛЖЕН автоматически блокироваться на шлюзе.

5. Проекция на плоские среды исполнения

Если целевая платформа не поддерживает иерархическую точечную структуру (Kubernetes Namespaces, облачные проекты, объекты СУБД, теги ресурсов), точечный разделитель заменяется на `-` или `_`. Порядок компонентов **от частного к общему** ДОЛЖЕН быть сохранён. Склеивание компонентов и изменение их порядка (вынос окружения в начало) ЗАПРЕЩЕНО.

При трансляции в плоские контейнеры компоненты `{node}` и `{slot}` опускаются: платформа самостоятельно управляет объектами внутри своего периметра.

5.1. Матрица трансляции

Платформа / тип ресурса	Разделитель	Канонический формат
Сетевые имена (DNS / FQDN)	`.`	`{node}.{slot}.{system}.{perimeter}.{env}.{dom`
Крипто-идентификатор (SPIFFE ID)	`/`	`spiffe://{trust-domain}/env/{env}/perimeter/{perimeter}/syste`
Kubernetes Namespace	`-`	`{system}-{perimeter}-{env}`
Облачный проект / Resource Group	`-`	`{perimeter}-{env}`
База данных / схема СУБД	`_`	`{system}_{perimeter}_{env}`
Пользователь / роль СУБД	`_`	`user_{system}_{perimeter}_{env}`

5.2. Kubernetes Namespaces

Пространство имён — проекция стенда: `{system}-{perimeter}-{env}`.

- `billing-payments-prod`
- `antifraud-risk-test`
- `vault-secrets-infrastructure`
- `pgcluster-platform-prod`

Инвертированные формы (~~prod-payments-billing~~) ЗАПРЕЩЕНЫ.

Кластер оркестрации — самостоятельный цифровой актив и стенд контура `platform` своего окружения (`kube1.platform.prod`), его worker-узлы именуются по 4.3 (`worker-11.main.kube1.platform.prod.example.ru`). Пространства имён потребителей — стенды других контуров, размещённые на кластере. Связь

«пространство имён → кластер» ведётся инвентаризацией платформы и НЕ ДОЛЖНА включаться в имя пространства имён. Пространство имён НЕ ДОЛЖНО содержать стенды разных окружений. Кластер НЕ ДОЛЖЕН размещать стенды разных окружений без изоляции на уровне платформы.

5.3. Облачные проекты (Cloud Projects / Resource Groups)

Логический контейнер ресурсов облачного провайдера — проекция контура: `{perimeter}-{env}`.

- `payments-prod`
- `pilot-stage`
- `cicd-infrastructure`

5.4. Базы данных и пользователи СУБД

Имена логических баз данных (схем) и системных учётных записей СУБД собираются с разделителем `_`: `{system}_{perimeter}_{env}` и `user_{system}_{perimeter}_{env}`.

- `billing_payments_prod`
- `antifraud_risk_test`
- `user_billing_payments_prod`

6. Идентификационные атрибуты

Каждый ИТ-ресурс независимо от платформы размещения (виртуальные машины, ресурсы облаков, поды и пространства имён Kubernetes, рабочие места) ДОЛЖЕН размечаться тремя обязательными тегами (ключами метаданных), дублирующими семантику имени для инвентаризации, сбора метрик и FinOps:

Тег	Значение
`envspec.io/env`	Идентификатор окружения: `dev`, `test`, `stage`, `prod`, `infrastructure`, `workplace`

Тег	Значение
`envspec.io/perimeter`	Идентификатор контура
`envspec.io/system`	Мнемокод цифрового актива из корпоративного реестра

Необязательные теги: `envspec.io/slot` — слот; `envspec.io/site` — код площадки (3.7).

Если платформа не допускает символы `/` и `.` в ключах, ключи ДОЛЖНЫ записываться как `envspec_env`, `envspec_perimeter`, `envspec_system` (`envspec_slot`, `envspec_site`).

Значения тегов ресурса, имя которого построено по 4.3 или разделу 5, ДОЛЖНЫ совпадать с соответствующими компонентами имени. Расхождение — нарушение стандарта.

> ⚠ **О СОВМЕСТИМОСТИ С LEGACY-СИСТЕМАМИ**
>
> Для унаследованных и архитектурно сложных систем, изменение сетевых имён которых невозможно без нарушения работоспособности ландшафта, действует приоритет маркировки над переименованием:
>
> 1. Переименование существующих сетевых имён (DNS/FQDN) legacy-хостов НЕ ЯВЛЯЕТСЯ требованием стандарта. Имена могут оставаться неизменными до планового вывода из эксплуатации или миграции.
> 2. Для приведения такой системы к соответствию стандарту ДОСТАТОЧНО разметить её тремя обязательными тегами на уровне платформы виртуализации или облака.
> 3. Корректные теги делают систему прозрачной для автоматизации и инвентаризации и полностью компенсируют унаследованную структуру её DNS-имени.

7. Критерии соответствия стандарту

Для заявления о соответствии спецификации **EnvSpec Naming 1.0.x** ИТ-инфраструктура организации или её обособленный ландшафт ДОЛЖНЫ

удовлетворять следующим автоматизированно проверяемым критериям:

1. **Полнота разметки.** 100 % активных вычислительных ресурсов, рабочих мест и контейнерных пространств (Namespaces) имеют заполненные обязательные теги `envspec.io/*`.
2. **Валидность идентификаторов.** 0 ресурсов со значением `envspec.io/env` вне списка 3.1. Идентификаторы контуров `platform` и `external` используются только по 3.4.
3. **Валидность DNS для новых ресурсов.** 100 % записей внутренних DNS-зон, созданных после даты принятия стандарта и указывающих на инфраструктурные узлы, соответствуют шаблону 4.3, включая обязательный компонент `{slot}`.
4. **Аудит legacy-систем.** Все системы, чьи DNS-имена не соответствуют шаблону 4.3, имеют 100 % покрытие обязательными тегами (раздел 6). Это признаётся полным соответствием стандарту.
5. **Чистота имён.** В именах новых ресурсов отсутствуют признаки версий ПО (кроме компонента `{slot}`), параметры SLA, классы критичности, коды площадок и роли сотрудников.
6. **Согласованность слоёв.** Для 100 % ресурсов, имеющих одновременно FQDN по 4.3, SPIFFE ID по 4.4 и обязательные теги, компоненты `env`, `perimeter`, `system` во всех трёх слоях совпадают.

Приложение A. Справочные регулярные выражения

Для линтеров, проверок в CI/CD и политик допуска (Validating Webhooks в Kubernetes):

Объект	Регулярное выражение
Компонент имени (RFC 1123, ≤ 63)	`^[a-z0-9]([a-z0-9-]{0,61}[a-z0-9])?$`
Базовый суффикс `{perimeter}`,	`^[a-z0-9]([a-z0-9-]{0,11}[a-z0-9])?$`

Объект	Регулярное выражение
`{system}` (≤ 13)	
Декомпозированный суффикс `{system}` (с дефисом, ≤ 16)	`^[a-z0-9]([a-z0-9-]{0,14}[a-z0-9])?$`
`{env}`	`^(dev|test|stage|prod|infrastructure|workplace)$`
FQDN узла (4.3, с учётом зоны)	`^[a-z0-9]([a-z0-9-]{0,61}[a-z0-9])?\.[a-z0-9]([a-z0-9-]9])?\.[a-z0-9]([a-z0-9-]{0,14}[a-z0-9])?\.[a-z0-9]([a-z0-9])?\.(dev|test|stage|prod|infrastructure|workplace)`
SPIFFE ID (4.4)	`^spiffe://[a-z0-9.-]+/env/(dev|test|stage|prod|infrastructure|workplacez0-9]([a-z0-9-]{0,11}[a-z0-9])?/system/[a-z0-9]([a-z0-9]9])?/slot/[a-z0-9]([a-z0-9-]{0,61}[a-z0-9])?$`
Kubernetes Namespace (5.2)	`^[a-z0-9]([a-z0-9-]{0,14}[a-z0-9])?-[a-z0-9]([a-z0-9-]{-(dev|test|stage|prod|infrastructure|workplace)$`
Облачный проект (5.3)	`^[a-z0-9]([a-z0-9-]{0,11}[a-z0-9])?-(dev|test|stage|prod|infrastructure|workplace)$`
База данных / схема (5.4)	`^[a-z0-9]([a-z0-9-]{0,14}[a-z0-9])?_[a-z0-9]([a-z0-9-]{_(dev|test|stage|prod|infrastructure|workplace)$`
Пользователь СУБД (5.4)	`^user_[a-z0-9]([a-z0-9-]{0,14}[a-z0-9])?_[a-z0-9]([a-z0z0-9])?_(dev|test|stage|prod|infrastructure|workplace)$`

Выражения для FQDN и SPIFFE ID проверяют формат и длины компонентов; допустимость конкретных значений `{perimeter}` и `{system}` проверяется по реестру цифровых активов и реестру контуров.

Об этом документе

Стандарт версионируется по SemVer: MAJOR — несовместимые изменения модели или правил, MINOR — новые правила и идентификаторы, PATCH — уточнения формулировок. Спецификация фиксирует правила именования, маркировки и криптографической идентичности. Процессы управления политиками безопасности и согласования доступов в её область не входят.

Автор — Ганюшкин Андрей Александрович (Andrei Ganiushkin). Текст спецификации распространяется по лицензии CC BY-SA 4.0: его разрешается свободно использовать, копировать, адаптировать и включать во внутренние регламенты, стандарты и политики организации с указанием авторства и распространением производных на тех же условиях.

Ограничение на использование имени. Запрещается использовать наименования «EnvSpec», «EnvSpec Naming», «EnvSpec Governance», «EnvSpec Security» и производные от них для публичного распространения изменённых версий данного документа. Публичные производные документы ДОЛЖНЫ быть полностью переименованы (например, «Стандарт именования ИТ-ресурсов Компании X»). Наименование EnvSpec в изменённых публичных документах допускается только в контексте цитирования первоисточника: «Создано на основе методологии EnvSpec за авторством Ганюшкина А.А.».