3 июня 2025 Москва, LOFT HALL#2

БЕКОН '25

Конференция по БЕзопасности
КОНтейнеров и контейнерных сред

Чеклист безопасности ML-кластеров

Панченко Николай

Руководитель направления рантайм защиты Т-Банк

LUNTRY

Т БАНК

Николай Панченко

Т БАНК

Руководитель направления рантайм защиты Т-Банк

О себе:

- Опыт в ИБ более 10 лет
- Докладчик конференций:
  - HighLoad++
  - PHDays
  - OffZone
  - БЕКОН и др.
- Убежден в том, что отделы ИБ и ИТ должны работать сообща для достижения наивысших результатов =)

TG: @Yours_rage

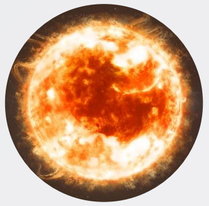
www.tbank.ru

План:

- Нужно ли строить отдельные кластеры для ML K8s?
- Подходы к подключению устройств в K8s
- Уровни безопасности ML K8s
- Угрозы и модель нарушителя для ML K8s
- Формируем чеклист для ML K8s
- Выводы

Нужно ли строить отдельные кластеры для ML K8s?

Разделение кластеров K8s по назначению



Пользовательские (User-space)

- Позволяют запускать пользовательские нагрузки
- Пользователи имеют доступ к прод-кластеру K8s (ну или “крутят” там свои нагрузки если у вас ZTP)
- Это кластеры, которые зарабатывают =)

Служебные (Service)

- Обеспечивают работу пользовательских кластеров
- Только администраторы имеют прямой доступ к кластеру
- Их меньше, но их тоже необходимо защищать!

https://vk.cc/cMvr9h – **“Использование единой платформы разработки для повышения безопасности K8s” VK K8s Conf 2023**

Виды пользовательских K8s кластеров

Общего назначения (Workload)

- Запуск пользовательского ПО
- Разделение компонентов User/Admin
- Обычно присутствуют все типы K8s ресурсов
- Могут присутствовать публикации

Кластеры-перекладчики (Batch)

- Запуск специального ПО (DATA): Apache Spark, Apache Flink, Quix и др.
- Перекладчики данных
- Исполнение Job, CronJob, Pods
- Как правило, публикаций нет =)

Кластеры для ML (ML-Core)

- Запуск специального ПО (ML): Apache AirFlow, KubeFlow, MLFlow
- Доступ к устройствам (NVIDIA)
- Запрет прямого доступа к K8s API
- Как правило, публикаций нет =)

https://vk.cc/cMvr9h – “Использование единой платформы разработки для повышения безопасности K8s” VK K8s Conf 2023

Виды пользовательских K8s кластеров

Общего назначения (Workload)

- Запуск пользовательского ПО
- Разделение компонентов User/Admin
- Обычно присутствуют все типы K8s ресурсов
- Могут присутствовать публикации

Кластеры-перекладчики (Batch)

- Запуск специального ПО (DATA): Apache Spark, Apache Flink, Quix и др.
- Перекладчики данных
- Исполнение Job, CronJob, Pods
- Как правило, публикаций нет =)

Кластеры для ML (ML-Core)

- Запуск специального ПО (ML): Apache AirFlow, KubeFlow, MLFlow
- Доступ к устройствам (NVIDIA)
- Запрет прямого доступа к K8s API
- Как правило, публикаций нет =)

https://vk.cc/cMvr9h – “Использование единой платформы разработки для повышения безопасности K8s” VK K8s Conf 2023

Нужно ли строить отдельные кластеры для ML K8s?

Плюсы

- Изоляция устройств хостовой ОС
- Снижение риска шумных соседей
- Получение возможности гибкой настройки кластера под задачу
- Улучшение безопасности и упрощение логирования событий
- Кастомные политики Policy engine и уникальный для ML харденинг
- Упрощение контроля доступа
- Упрощение процесса обновлений Workload кластеров (не завязаны на версию драйверов для nVidia)

Минусы

- Увеличение затрат на физические ресурсы (дублирование мастер-нод)
- Нужны дополнительные человеческие ресурсы на поддержку
- Необходимо получение выделенных доступов до хранилищ и интернета
- Повышение риска низкой утилизации при отсутствии ML-задач
- Сложность масштабирования под разные “плавающие” нагрузки
- Запрещено публиковать сервисы =)

Нужно ли строить отдельные кластеры для ML K8s?

Да, если в этом есть необходимость =)

Виды пользовательских K8s кластеров

Общего назначения (Workload)

- Запуск пользовательского ПО
- Разделение компонентов User/Admin
- Обычно присутствуют все типы K8s ресурсов
- Могут присутствовать публикации

Кластеры-перекладчики (Batch)

- Запуск специального ПО (DATA): Apache Spark, Apache Flink, Quix и др.
- Перекладчики данных
- Исполнение Job, CronJob, Pods
- Как правило, публикаций нет =)

Кластеры для ML (ML-Core)

- Запуск специального ПО (ML): Apache AirFlow, KubeFlow, MLFlow
- Доступ к устройствам (NVIDIA)
- Запрет прямого доступа к K8s API
- Как правило, публикаций нет =)

https://vk.cc/cMvr9h – “Использование единой платформы разработки для повышения безопасности K8s” VK K8s Conf 2023

K8s devices

K8s device (устройство) - любое аппаратное, физическое или виртуальное устройство, которое можно использовать внутри контейнеров для выполнения специализированных вычислений или обработки данных.

K8s devices

K8s device (устройство) - любое аппаратное, физическое или виртуальное устройство, которое можно использовать внутри контейнеров для выполнения специализированных вычислений или обработки данных.

Примеры:

- GPU;
- TPU;
- FPGA;
- DPU;
- SmartNIC;
- ASIC;
- NVMe-диски;
- криптографические ускорители;
- и др.

Подходы к подключению устройств в K8s

Подходы к подключению устройств в K8s

Кто какие знает...

Подходы к подключению устройств в K8s

Кто какие знает...

spec.template.spec.containers.{name}
.securityContext.privileged: true

+

Mount => hostPath.path: /

Подходы к подключению устройств в K8s

Кто какие знает...

spec.template.spec.containers.{name}
.securityContext.privileged: true

+

Mount => hostPath.path: /

Подходы к подключению устройств в K8s

Device Plugin

CDI

DRA

Подходы к подключению устройств в K8s

Device Plugin

CDI

DRA

K8s Device Plugin

K8s Device Plugin — это механизм, позволяющий Kubernetes управлять специализированными аппаратными устройствами (GPU, NICs, QAT, NVMe и др.).

* Ссылка на документацию K8s - https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/device-plugins/

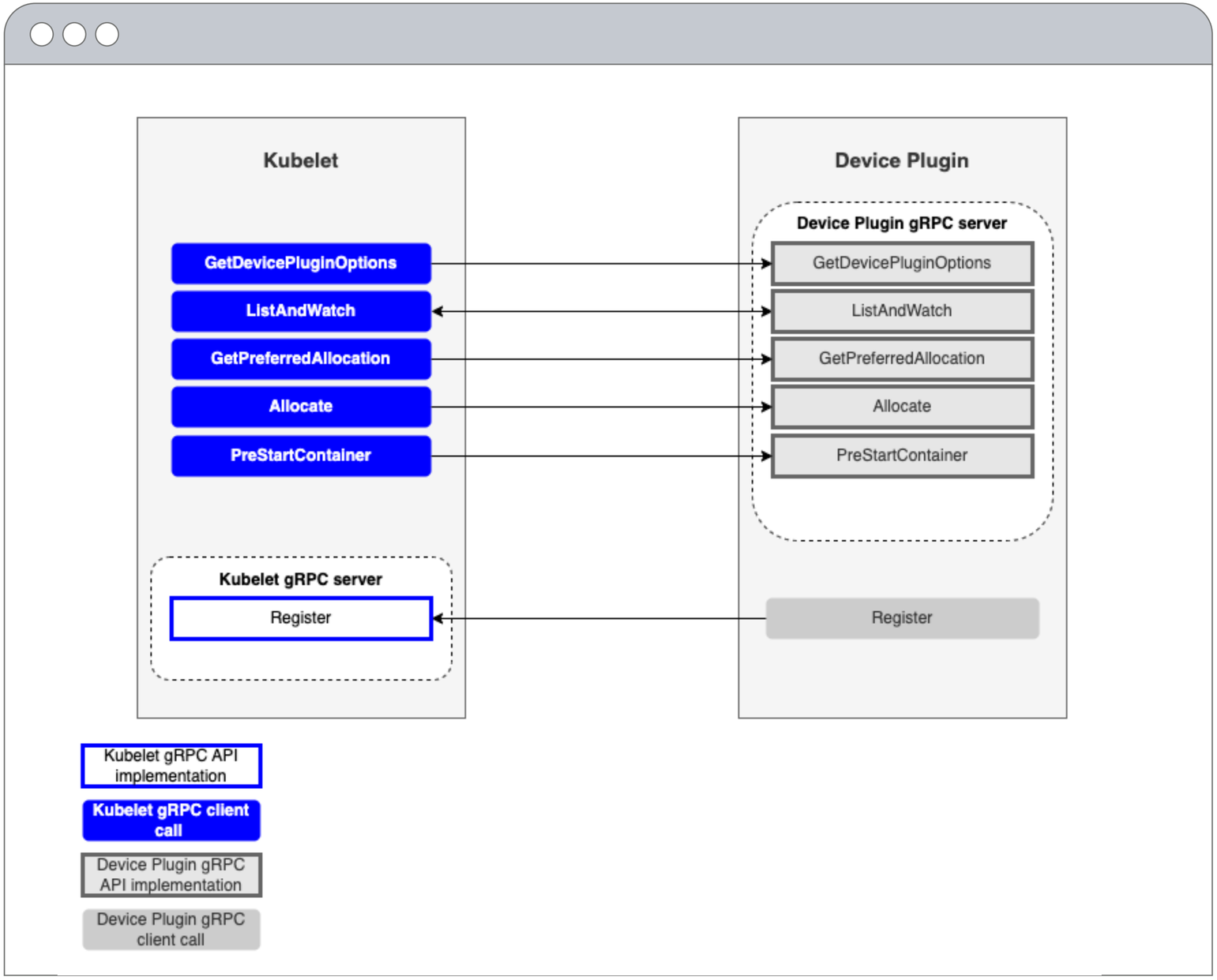
K8s Device Plugin

K8s Device Plugin — это механизм, позволяющий Kubernetes управлять специализированными аппаратными устройствами (GPU, NICs, QAT, NVMe и др.).

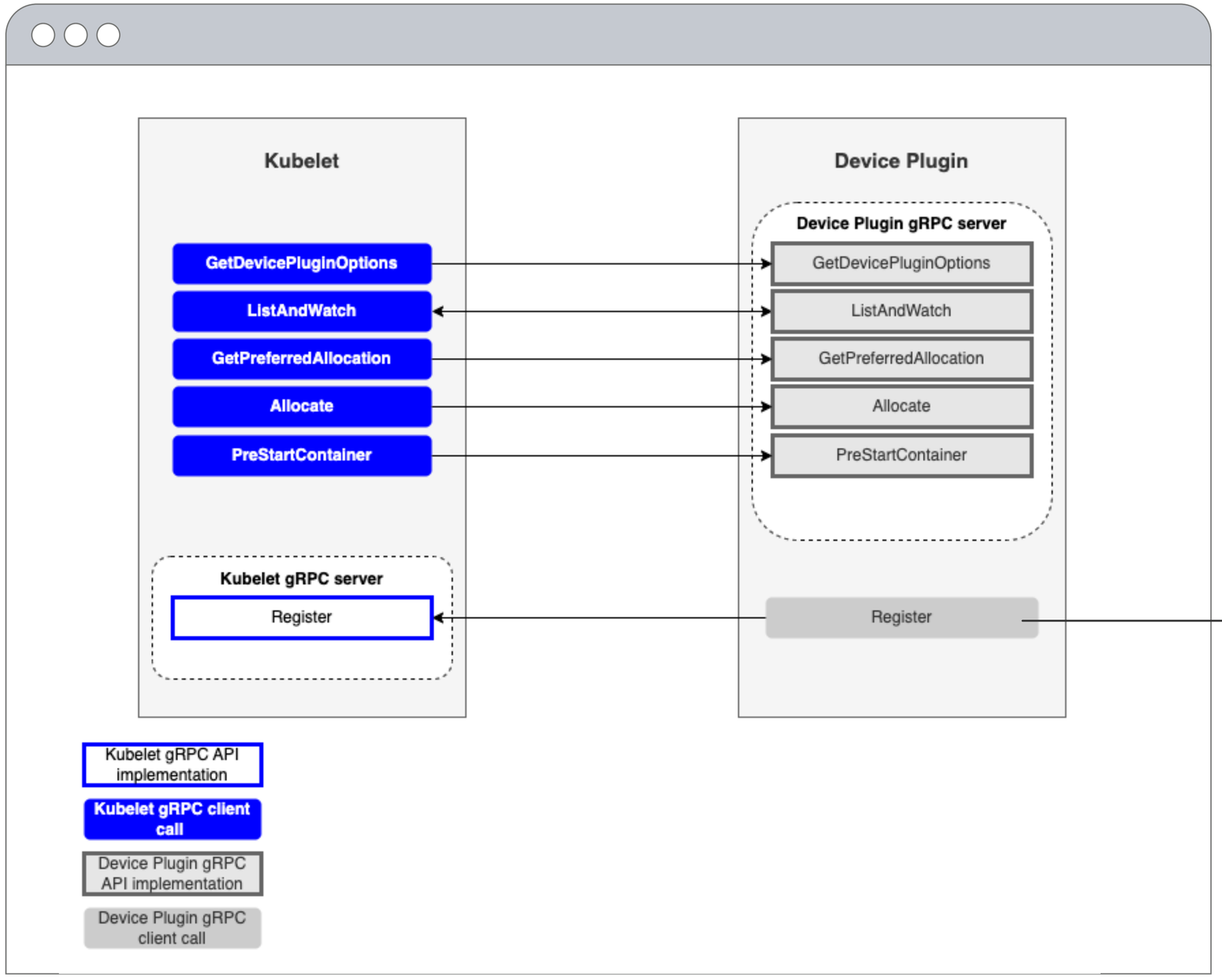
- Появился в stable с c1.26
- Работает как дополнительный процесс на Worker-ноде
- Общается с Kubelet предоставляет информацию о доступных устройствах
- Работает по GRPC

* Ссылка на документацию K8s - https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/device-plugins/

K8s Device Plugin

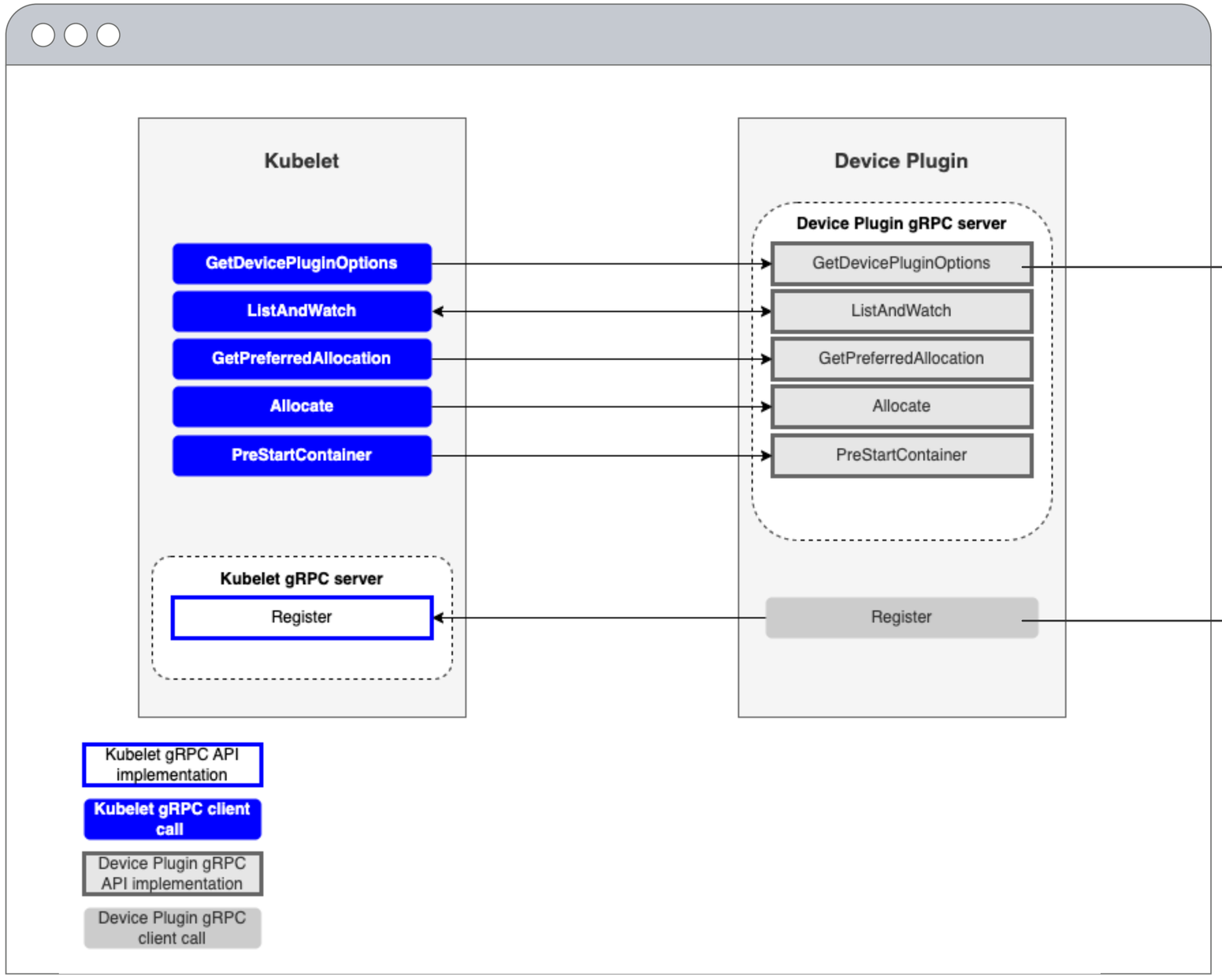


K8s Device Plugin



Регистрация Device Plugin, предоставление интерфейса для управления устройствами

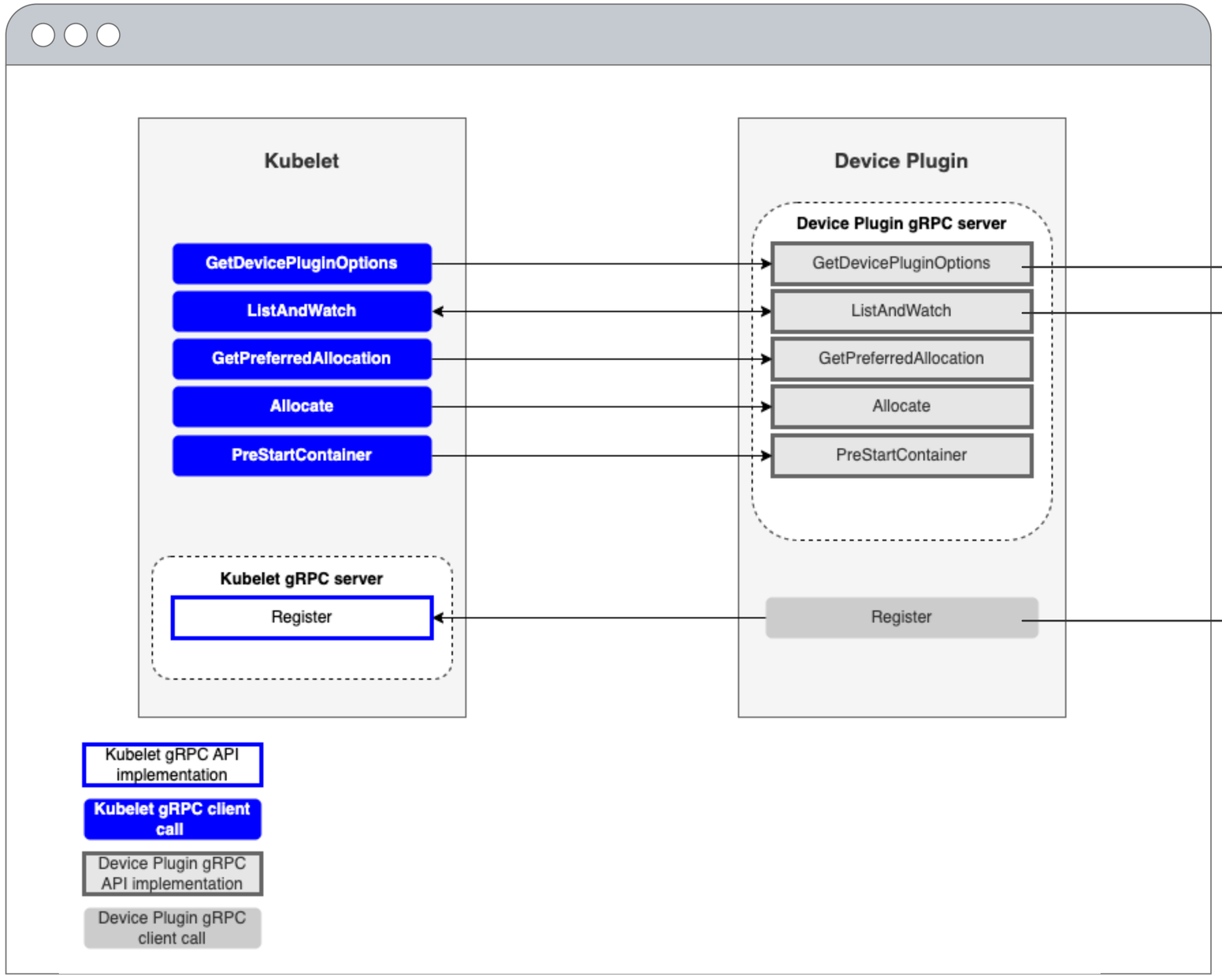
K8s Device Plugin



Доступные опции

Регистрация Device Plugin, предоставление интерфейса для управления устройствами

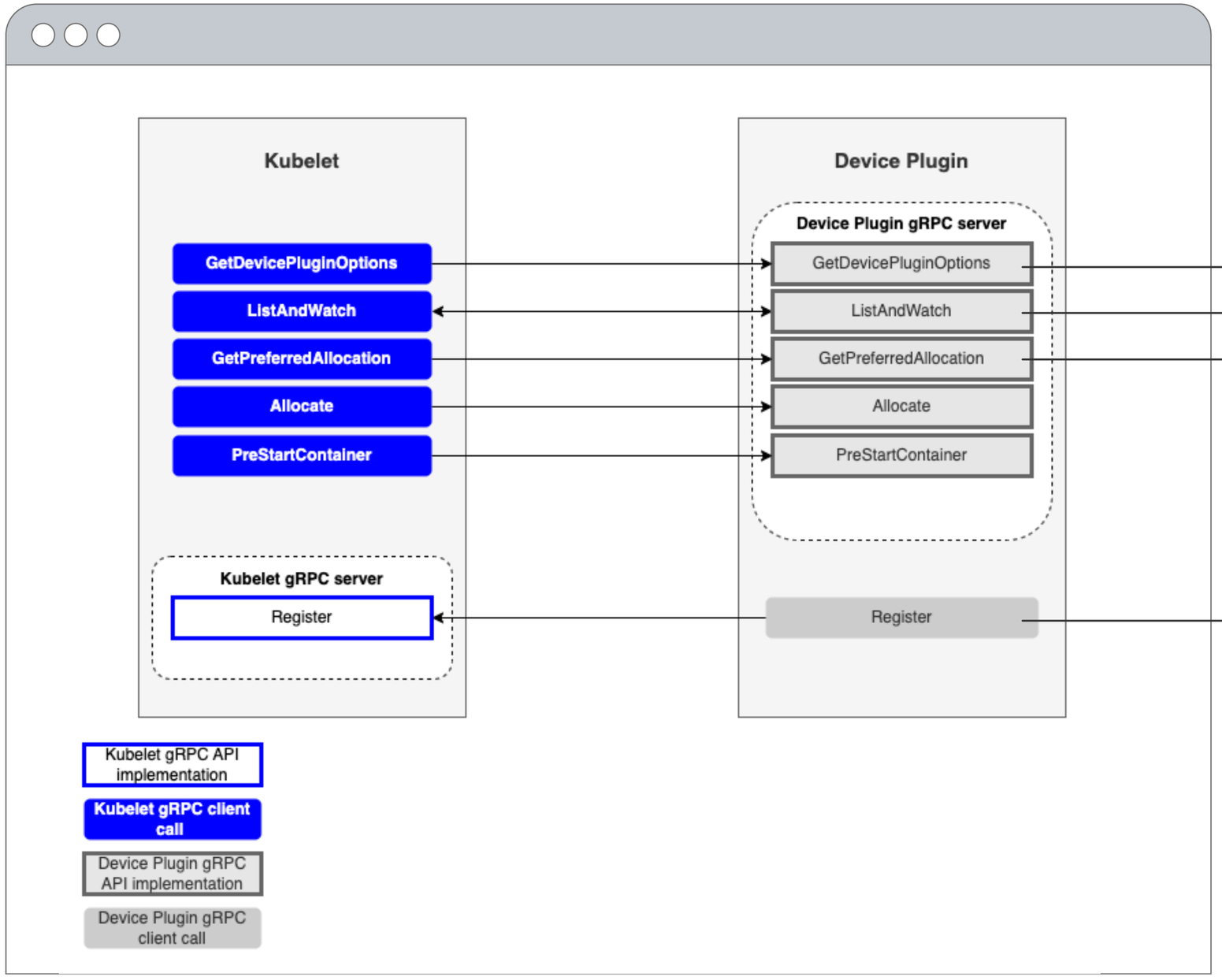
K8s Device Plugin



Доступные опции
Доступные устройства

Регистрация Device Plugin,
предоставление интерфейса
для управления устройствами

K8s Device Plugin



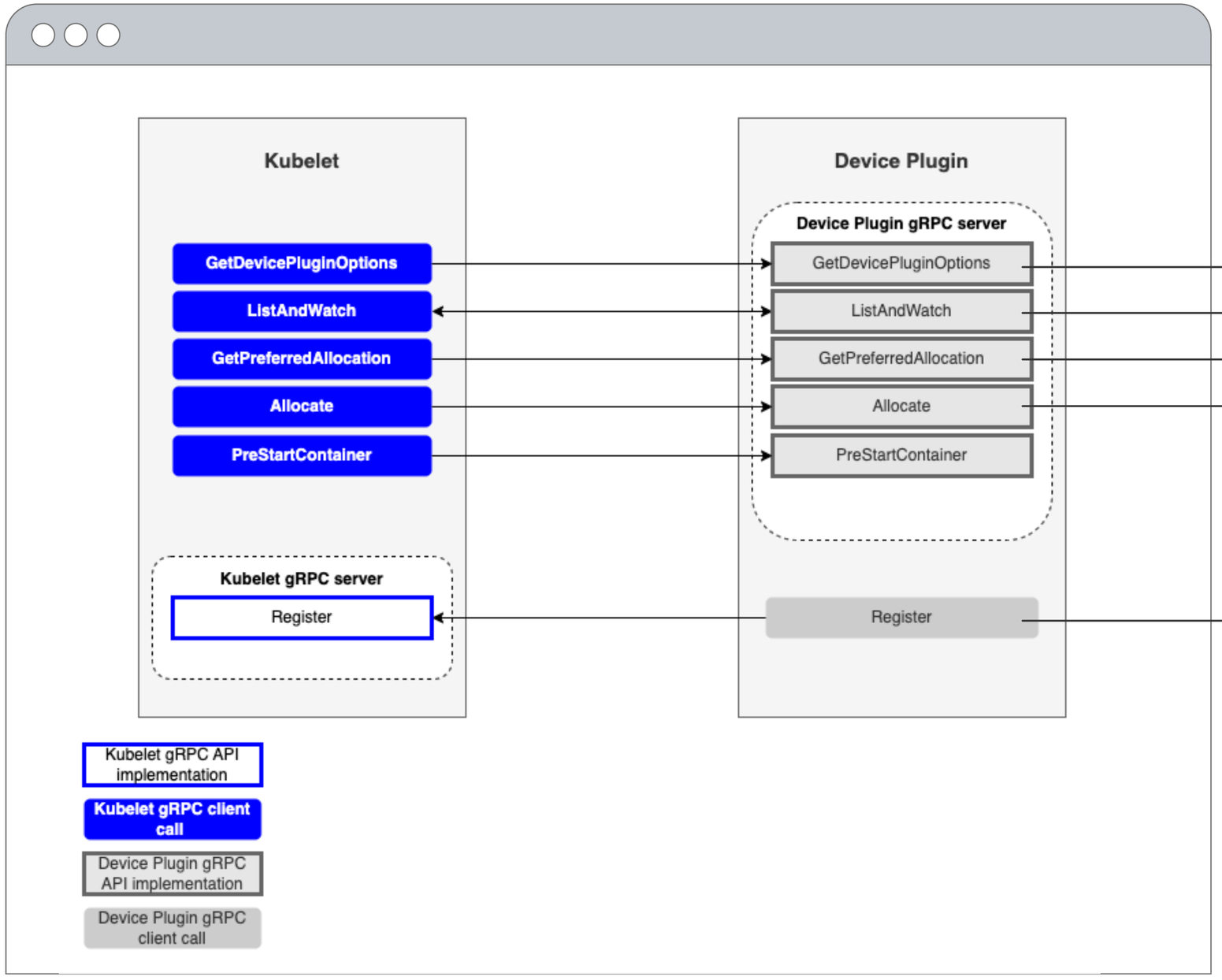
Доступные опции

Доступные устройства

Выбор подходящего устройства

Регистрация Device Plugin, предоставление интерфейса для управления устройствами

K8s Device Plugin



Доступные опции

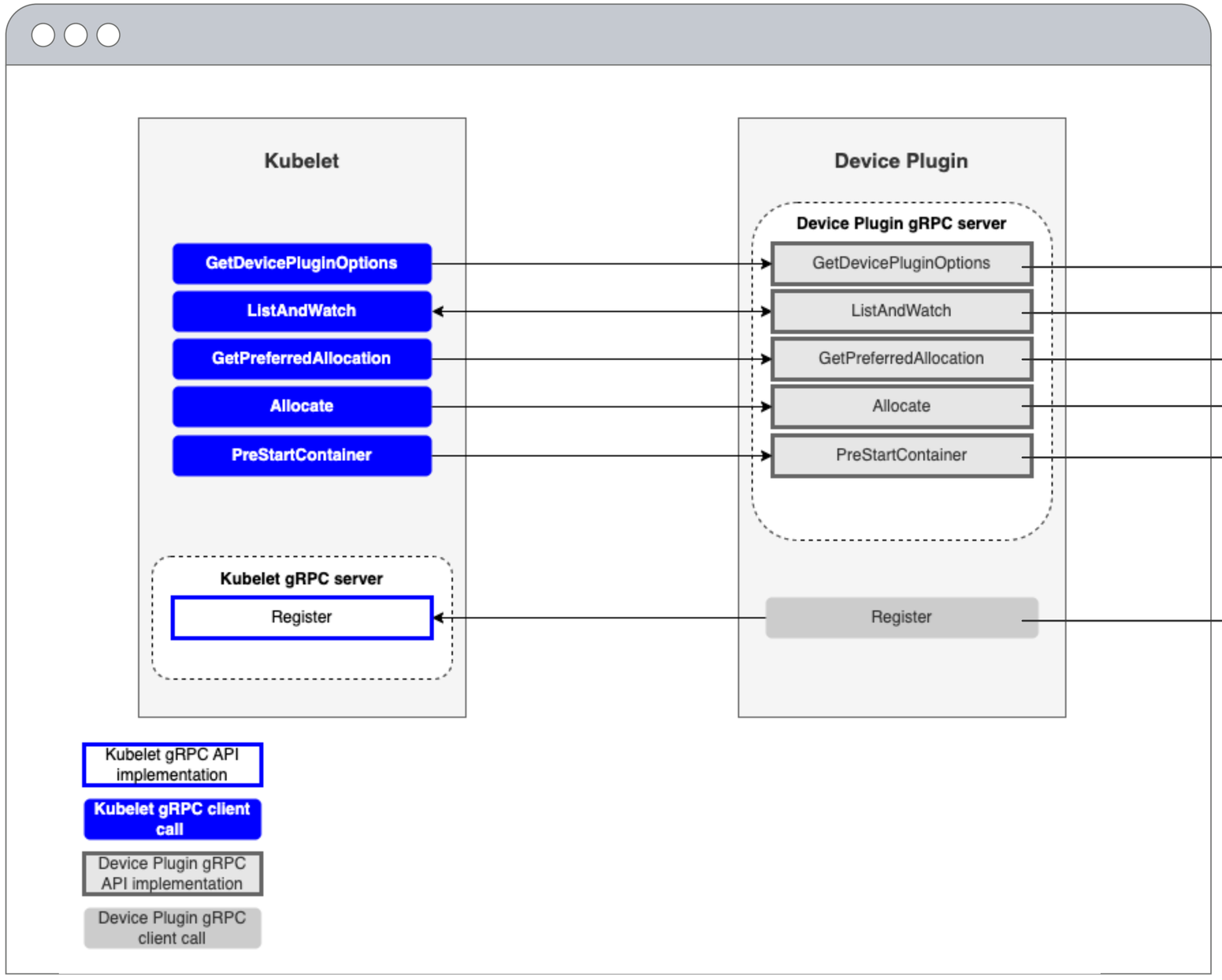
Доступные устройства

Выбор подходящего устройства

Выделение ресурсов

Регистрация Device Plugin, предоставление интерфейса для управления устройствами

K8s Device Plugin



Доступные опции

Доступные устройства

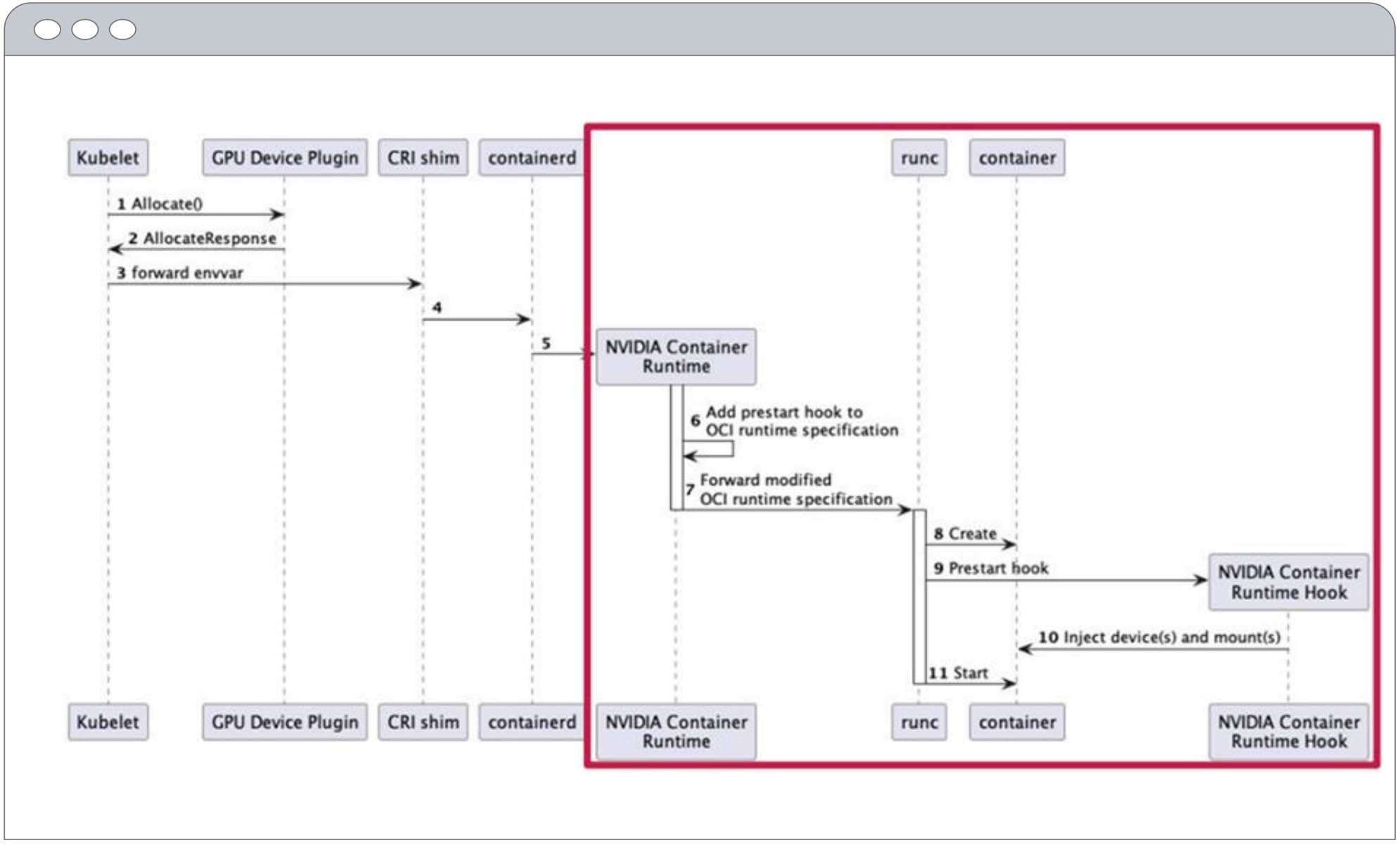
Выбор подходящего устройства

Выделение ресурсов

Вызов получения «доп. опций» перед запуском контейнера

Регистрация Device Plugin, предоставление интерфейса для управления устройствами

Workflow Device Plugin (NVidia GPU Operator)



Подходы к подключению устройств в K8s

Device Plugin

CDI

DRA

K8s CDI

K8s CDI (Container Device Interface) — это спецификация для container runtime, необходимая для поддержки устройств сторонних производителей.

* Ссылка на репозиторий CDI - https://github.com/cncf-tags/container-device-interface

K8s CDI

K8s CDI (Container Device Interface) — это спецификация для container runtime, необходимая для поддержки устройств сторонних производителей.

- Спецификация CDI основана на спецификации CNI (в alpha с K8s v1.27)
- Вводит абстрактное понятие устройства как ресурса `vendor.com/class=unique_name`
- Работает через FQDN и CDI File с настройками от вендора
- CDI не управляет ресурсами самостоятельно (управляет оркестратор)

* Ссылка на репозиторий CDI - https://github.com/cncf-tags/container-device-interface

K8s CDI спецификация для NVIDIA GPU Operator

```yaml
---
cdiVersion: 0.5.0
kind: nvidia.com/gpu
devices:
- name: gpu0
  containerEdits:
   deviceNodes:
   - path: /dev/nvidia0
   - path: /dev/nvidiactl
   mounts:
   - containerPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
     hostPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
     options: [ro, nosuid, nodev, bind]
   - containerPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
     hostPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
     options: [ro, nosuid, nodev, bind]
   - containerPath: /usr/bin/nvidia-smi
     hostPath: /usr/bin/nvidia-smi
     options: [ro, nosuid, nodev, bind]
   hooks:
   - hookName: createContainer
     path: /usr/bin/nvidia-ctk
     args:
     - /usr/bin/nvidia-ctk
     - hook
     - update-ldcache
     - --folder
     - /usr/lib/x86_64-linux-gnu
```

K8s CDI спецификация для NVIDIA GPU Operator

FQDN

```
---
cdiVersion: 0.5.0
kind: nvidia.com/gpu
devices:
- name: gpu0
  containerEdits:
    deviceNodes:
    - path: /dev/nvidia0
    - path: /dev/nvidiactl
    mounts:
    - containerPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
      hostPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
      options: [ro, nosuid, nodev, bind]
    - containerPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
      hostPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
      options: [ro, nosuid, nodev, bind]
    - containerPath: /usr/bin/nvidia-smi
      hostPath: /usr/bin/nvidia-smi
      options: [ro, nosuid, nodev, bind]
    hooks:
    - hookName: createContainer
      path: /usr/bin/nvidia-ctk
      args:
      - /usr/bin/nvidia-ctk
      - hook
      - update-ldcache
      - --folder
      - /usr/lib/x86_64-linux-gnu
```

K8s CDI спецификация для NVIDIA GPU Operator

FQDN

↓

NODES

```
---
cdiVersion: 0.5.0
kind: nvidia.com/gpu
devices:
- name: gpu0
  containerEdits:
    deviceNodes:
    - path: /dev/nvidia0
    - path: /dev/nvidiactl
    mounts:
    - containerPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
      hostPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
      options: [ro, nosuid, nodev, bind]
    - containerPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
      hostPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
      options: [ro, nosuid, nodev, bind]
    - containerPath: /usr/bin/nvidia-smi
      hostPath: /usr/bin/nvidia-smi
      options: [ro, nosuid, nodev, bind]
    hooks:
    - hookName: createContainer
      path: /usr/bin/nvidia-ctk
      args:
      - /usr/bin/nvidia-ctk
      - hook
      - update-ldcache
      - --folder
      - /usr/lib/x86_64-linux-gnu
```

K8s CDI спецификация для NVIDIA GPU Operator

FQDN

↓

NODES

↓

MOUNTS

```
---
cdiVersion: 0.5.0
kind: nvidia.com/gpu
devices:
- name: gpu0
  containerEdits:
   deviceNodes:
   - path: /dev/nvidia0
   - path: /dev/nvidiactl
   mounts:
   - containerPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
     hostPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
     options: [ro, nosuid, nodev, bind]
   - containerPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
     hostPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
     options: [ro, nosuid, nodev, bind]
   - containerPath: /usr/bin/nvidia-smi
     hostPath: /usr/bin/nvidia-smi
     options: [ro, nosuid, nodev, bind]
   hooks:
   - hookName: createContainer
     path: /usr/bin/nvidia-ctk
     args:
     - /usr/bin/nvidia-ctk
     - hook
     - update-ldcache
     - --folder
     - /usr/lib/x86_64-linux-gnu
```

K8s CDI спецификация для NVIDIA GPU Operator

FQDN

⬇

NODES

⬇

MOUNTS

⬇

HOOKS

```
---
cdiVersion: 0.5.0
kind: nvidia.com/gpu
devices:
- name: gpu0
  containerEdits:
   deviceNodes:
   - path: /dev/nvidia0
   - path: /dev/nvidiactl
   mounts:
   - containerPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
     hostPath: /usr/lib/x86_64-linux-gnu/libnvidia-ml.so.525.91.03
     options: [ro, nosuid, nodev, bind]
   - containerPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
     hostPath: /usr/lib/x86_64-linux-gnu/libcuda.so.525.91.03
     options: [ro, nosuid, nodev, bind]
   - containerPath: /usr/bin/nvidia-smi
     hostPath: /usr/bin/nvidia-smi
     options: [ro, nosuid, nodev, bind]
   hooks:
   - hookName: createContainer
     path: /usr/bin/nvidia-ctk
     args:
     - /usr/bin/nvidia-ctk
     - hook
     - update-ldcache
     - --folder
     - /usr/lib/x86_64-linux-gnu
```

Подходы к подключению устройств в K8s

Device Plugin

CDI

DRA

K8s DRA

K8s Dynamic Resource Allocation (DRA) - это новый API для запроса ресурсов в K8s позволяющий более гибко и эффективно распределять ресурсы, такие как графические процессоры или сетевые устройства, между рабочими нагрузками.

* Ссылка на документацию K8s - https://kubernetes.io/docs/concepts/scheduling-eviction/dynamic-resource-allocation/

K8s DRA

K8s Dynamic Resource Allocation (DRA) - это новый API для запроса ресурсов в K8s позволяющий более гибко и эффективно распределять ресурсы, такие как графические процессоры или сетевые устройства, между рабочими нагрузками.

- FEATURE STATE: Kubernetes v1.32 [beta] (enabled by default: false)
- DRA является логическим продолжением CDI
- Вводит DeviceClass, ResourceClaim, ResourceClaimTemplates и ResourceSlice
- v1.26 - v1.31 alpha «classic DRA» - больше не поддерживается

* Ссылка на документацию K8s - https://kubernetes.io/docs/concepts/scheduling-eviction/dynamic-resource-allocation/

K8s DRA DeviceClass

```yaml
apiVersion: resource.k8s.io/v1beta2
kind: DeviceClass
metadata:
  name: resource.example.com
spec:
  selectors:
  - cel:
      expression: device.driver == "resource-driver.example.com"
```

K8s DRA Create ResourceClaimTemplate

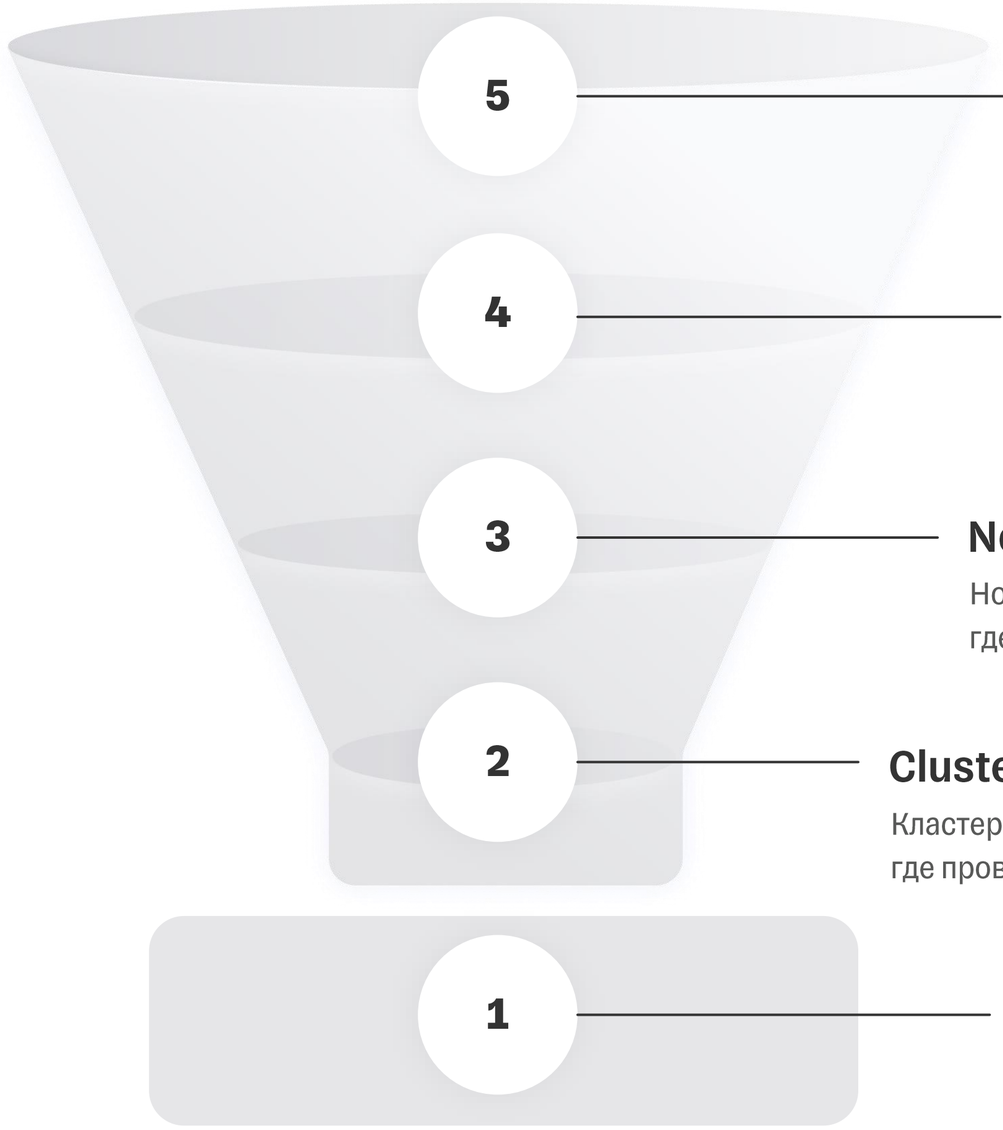
```yaml
apiVersion: resource.k8s.io/v1beta2
kind: ResourceClaimTemplate
metadata:
  name: large-black-cat-claim-template
spec:
  spec:
    devices:
      requests:
      - name: req-0
        exactly:
          deviceClassName: resource.example.com
          selectors:
          - cel:
              expression: |-
                device.attributes["resource-driver.example.com"].color == "black" &&
                device.attributes["resource-driver.example.com"].size == "large"
```

K8s DRA Create Pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-cats
spec:
  containers:
  - name: container0
    image: ubuntu:20.04
    command: ["sleep", "9999"]
    resources:
      claims:
      - name: cat-0
  - name: container1
    image: ubuntu:20.04
    command: ["sleep", "9999"]
    resources:
      claims:
      - name: cat-1
  resourceClaims:
  - name: cat-0
    resourceClaimTemplateName: large-black-cat-claim-template
  - name: cat-1
    resourceClaimTemplateName: large-black-cat-claim-template
```

Уровни безопасности ML K8s

Уровни безопасности ML K8s



На слайде представлена информация из закрытого исследования, проведенного Т-Банк совместно с компанией ООО “КлаудРан”

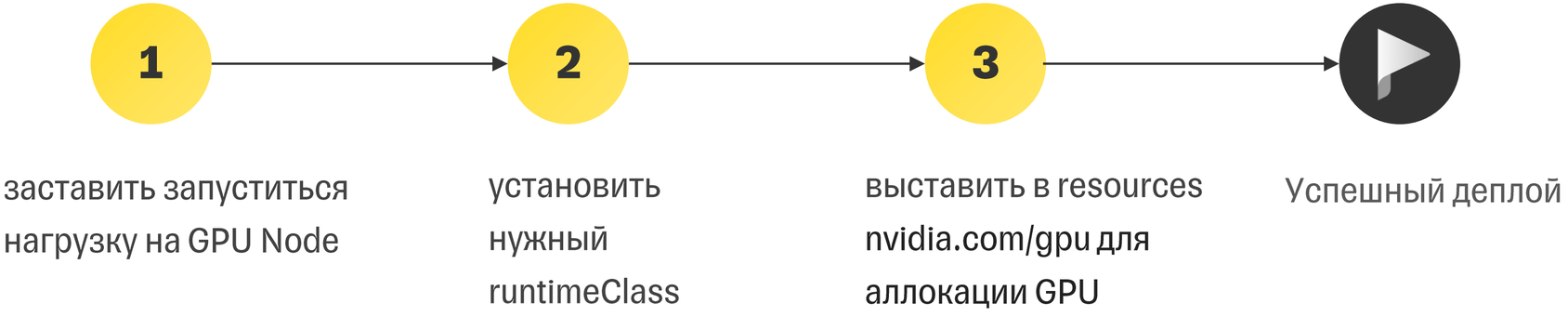
Угрозы и модель нарушителя для ML K8s

ML K8s модель нарушителя

модель нарушителя – пользователь системы с возможностью влиять на некоторые поля в манифесте создаваемой нагрузки.

ML K8s модель нарушителя

модель нарушителя – пользователь системы с возможностью влиять на некоторые поля в манифесте создаваемой нагрузки.



ML K8s модель угроз

Кража данных

01

Искажение данных

02

Уничтожение данных

03

Отказ в обслуживании

04

Нецелевое использование ресурсов

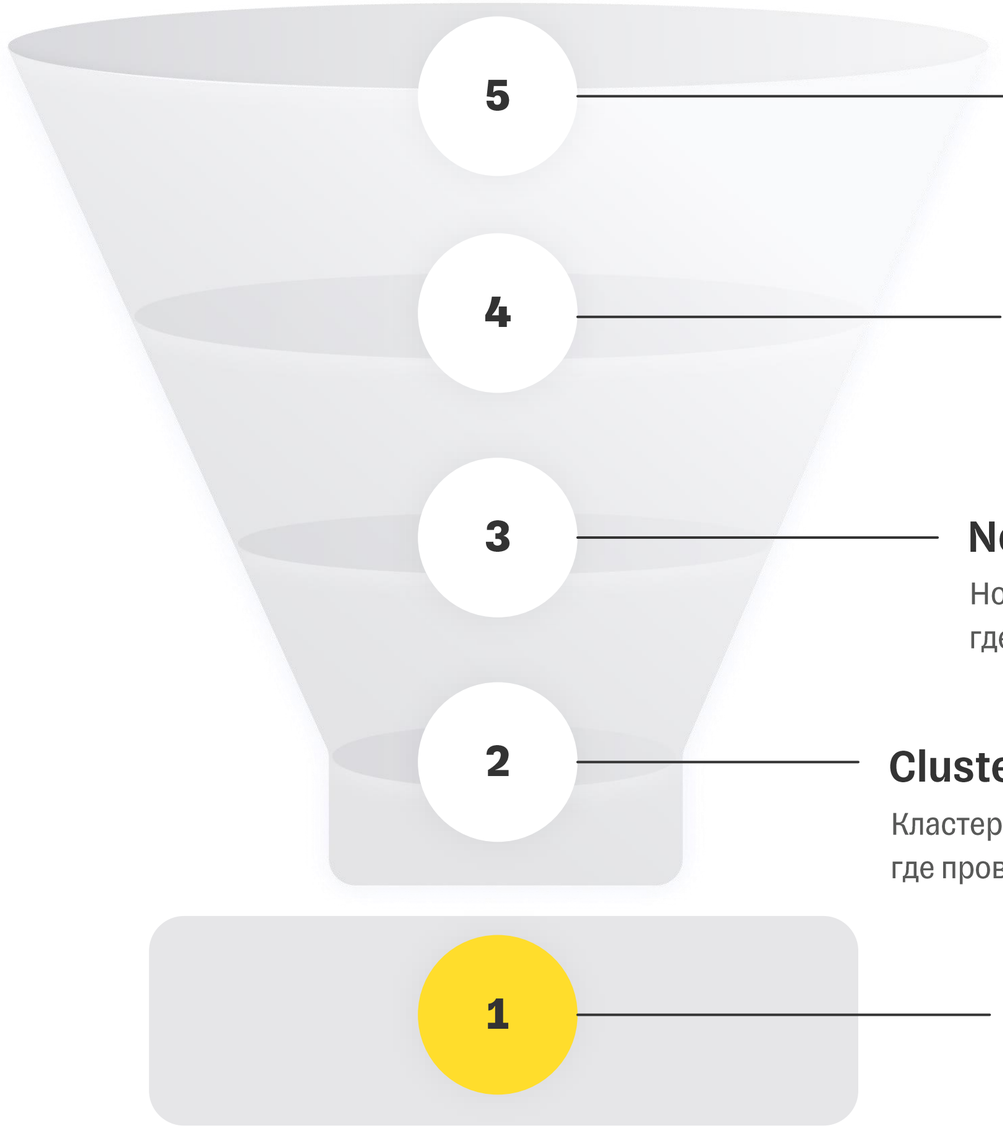
05

Финансовые потери

06

Формируем чеклист для ML K8s

Уровни безопасности ML K8s

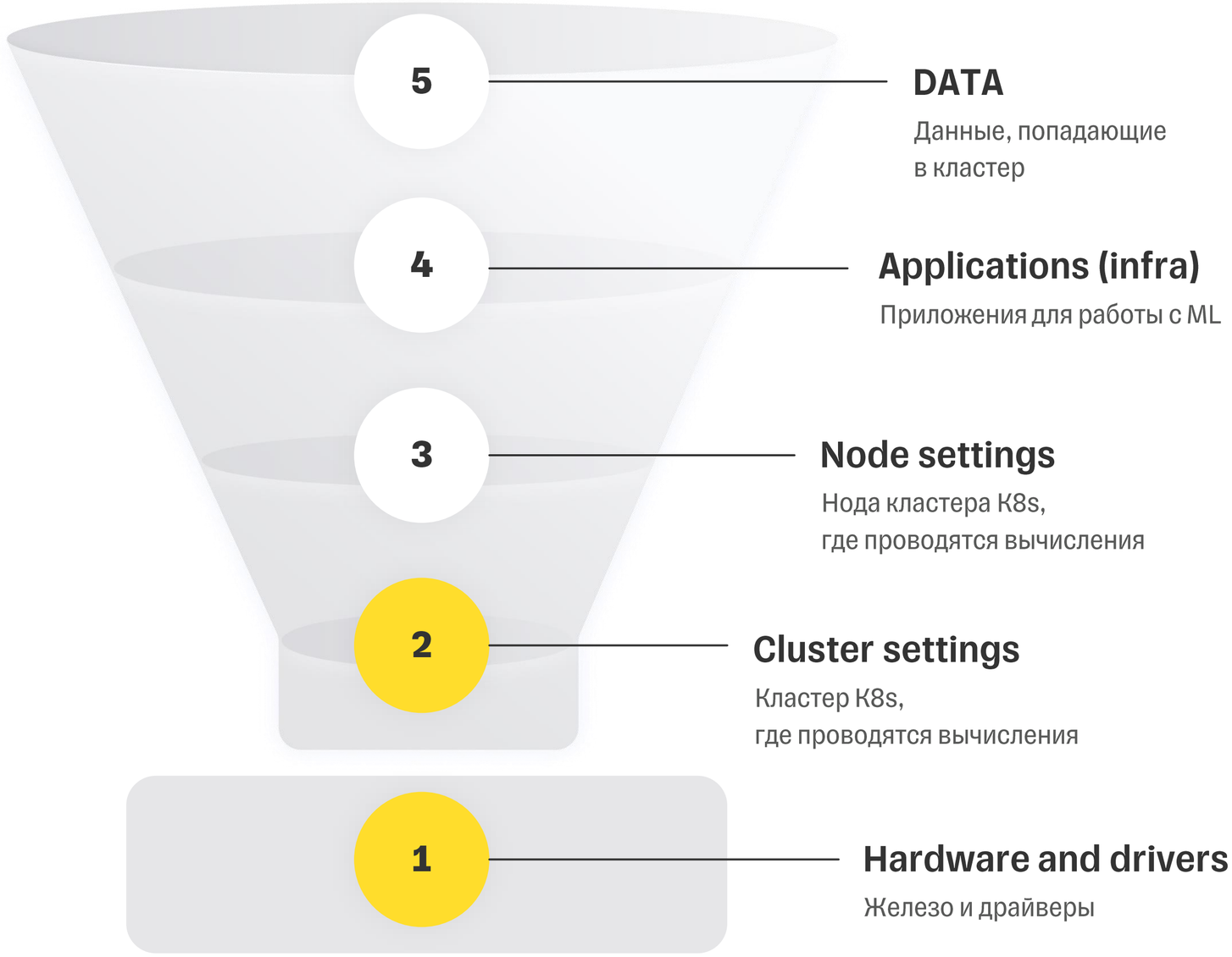


На слайде представлена информация из закрытого исследования, проведенного Т-Банк совместно с компанией ООО “КлаудРан”

Hardware and drivers

- Выявить текущий механизм подключения GPU устройств в кластер, проверить наличие использования CDI, если CDI нет – включить!
- Проверить версию драйвера и компонентов (NVIDIA Container Toolkit, libnvidia-container, NVIDIA GPU Operator при наличии)
- Проверить настройки config.toml для nvidia-container-runtime, чтобы убедиться, что ldconfig настроен на использование бинарного файла хоста: ldconfig = "/sbin/ldconfig"
- Если используется только Device Plugin, проверить наличие и настроить GPU Operator, чтобы предотвратить обход API Device Plugin с помощью переменной NVIDIA_VISIBLE_DEVICES
- Проверить CVE-2024-0132, CVE-2024-0134, CVE-2025-23359

Уровни безопасности ML K8s



На слайде представлена информация из закрытого исследования, проведенного Т-Банк совместно с компанией ООО “КлаудРан”

Cluster settings: шедулинг нагрузок

- Проверить механизмы шедулинга – labels, tolerations, affinity и anti affinity
- Проконтролировать, что нагрузки с использованием GPU попадут исключительно на Nodes с GPU
- Проконтролировать, что нагрузки без использования GPU не попадут на Nodes с GPU или попадут, но не смогут использовать проброс драйвера
- Проверить настройки политик Policy Engine и убедиться, что для всех типов нагрузок (Pod, Jobs etc.) запрещено вручную устанавливать node selector, tolerations, affinity, anti-affinity и node name

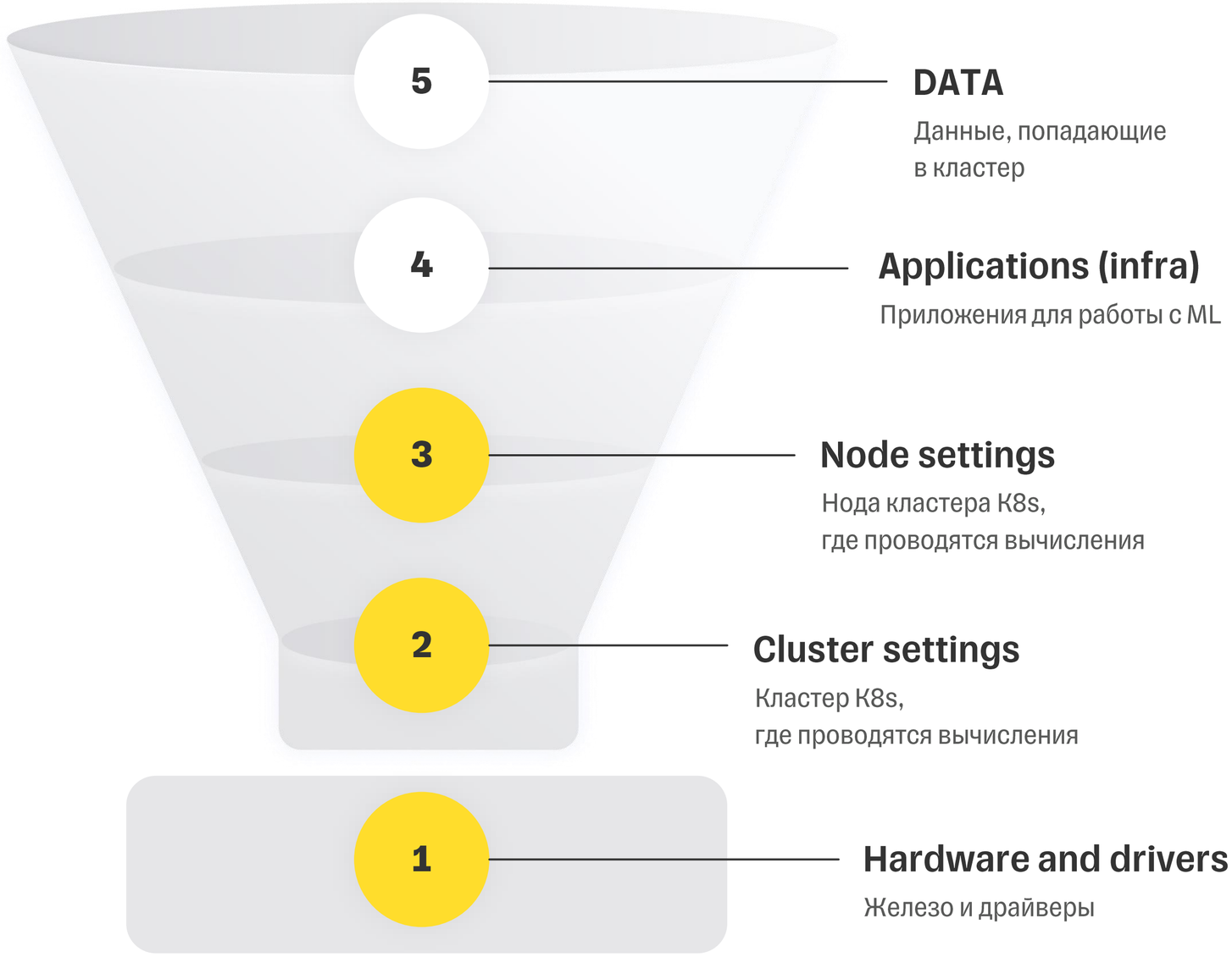
Cluster settings: общие настройки

- Проверить включенность Policy Engine в режиме enforce =)
- Проверить включенные сетевые политики в кластере
- Проконтролировать, что только нагрузки с определенным лейблом имеют доступ до прокси в Интернет
- Проверить доступность Vault для всех нагрузок в режиме изоляции

Cluster settings: общие настройки

- Проверить RBAC на наличие аномальных прав
- Проверить работоспособность системы мониторинга K8s (AuditLog)
- Образы для ML выкачиваются только из доверенного репозитория
- Проверить работоспособность комплексных средств защиты K8s (при наличии)

Уровни безопасности ML K8s

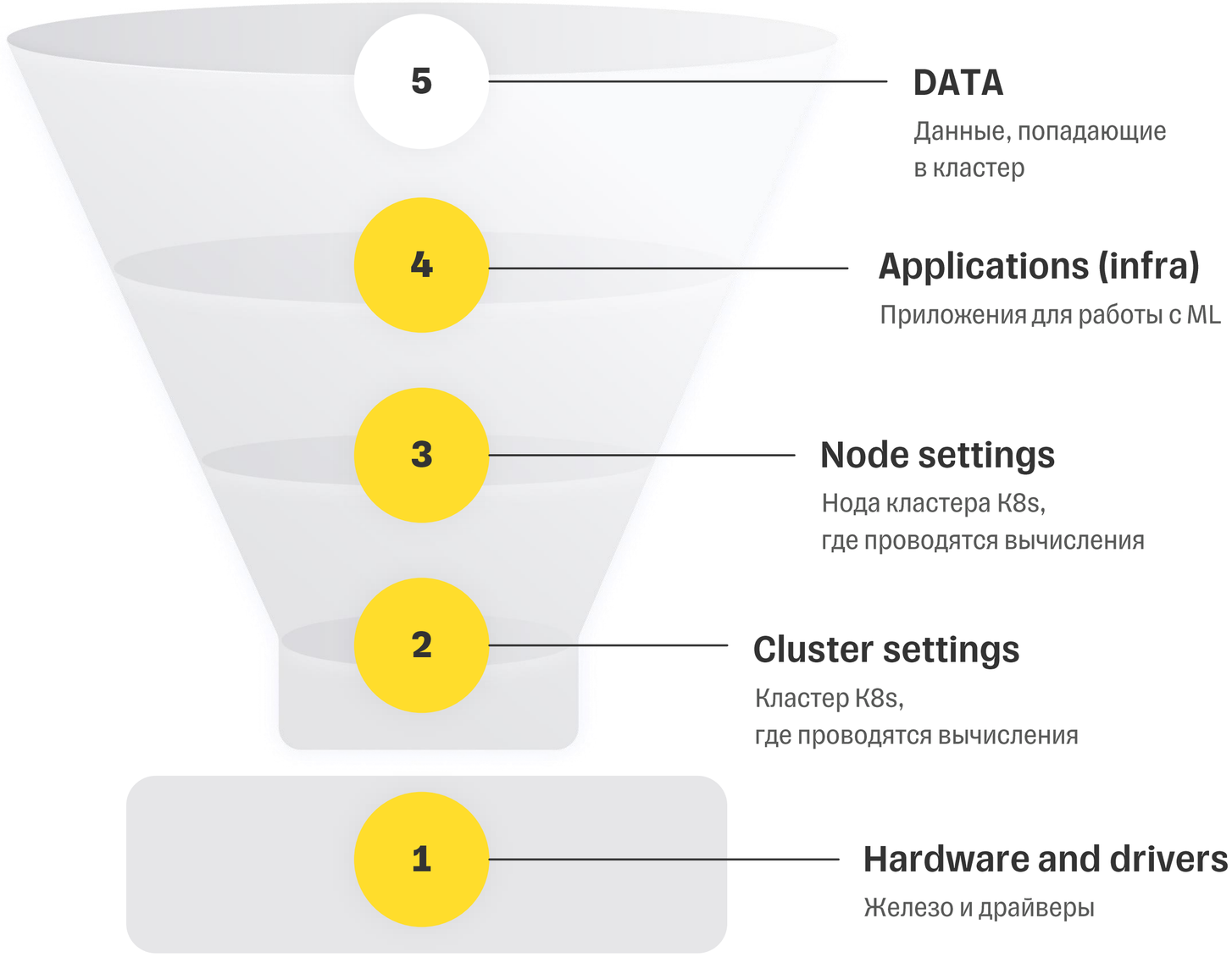


На слайде представлена информация из закрытого исследования, проведенного Т-Банк совместно с компанией ООО “КлаудРан”

Node settings

- Проверить назначенные для каждой ноды необходимые labels, tolerations, affinity и anti affinity => нагрузка с GPU должна уметь шедулиться на выбранную Node с GPU только согласно выставленной разметки
- Проконтролировать, что нагрузка с GPU шедулится на выбранную Node с GPU согласно квоте
- Проконтролировать, что на Node с GPU нет в данный момент нагрузок без использования GPU или есть, но нет проброса драйвера
- Проверить размещение необходимых компонентов на Node с GPU для запуска нагрузок с GPU (Device Plugin DaemonSet)

Уровни безопасности ML K8s

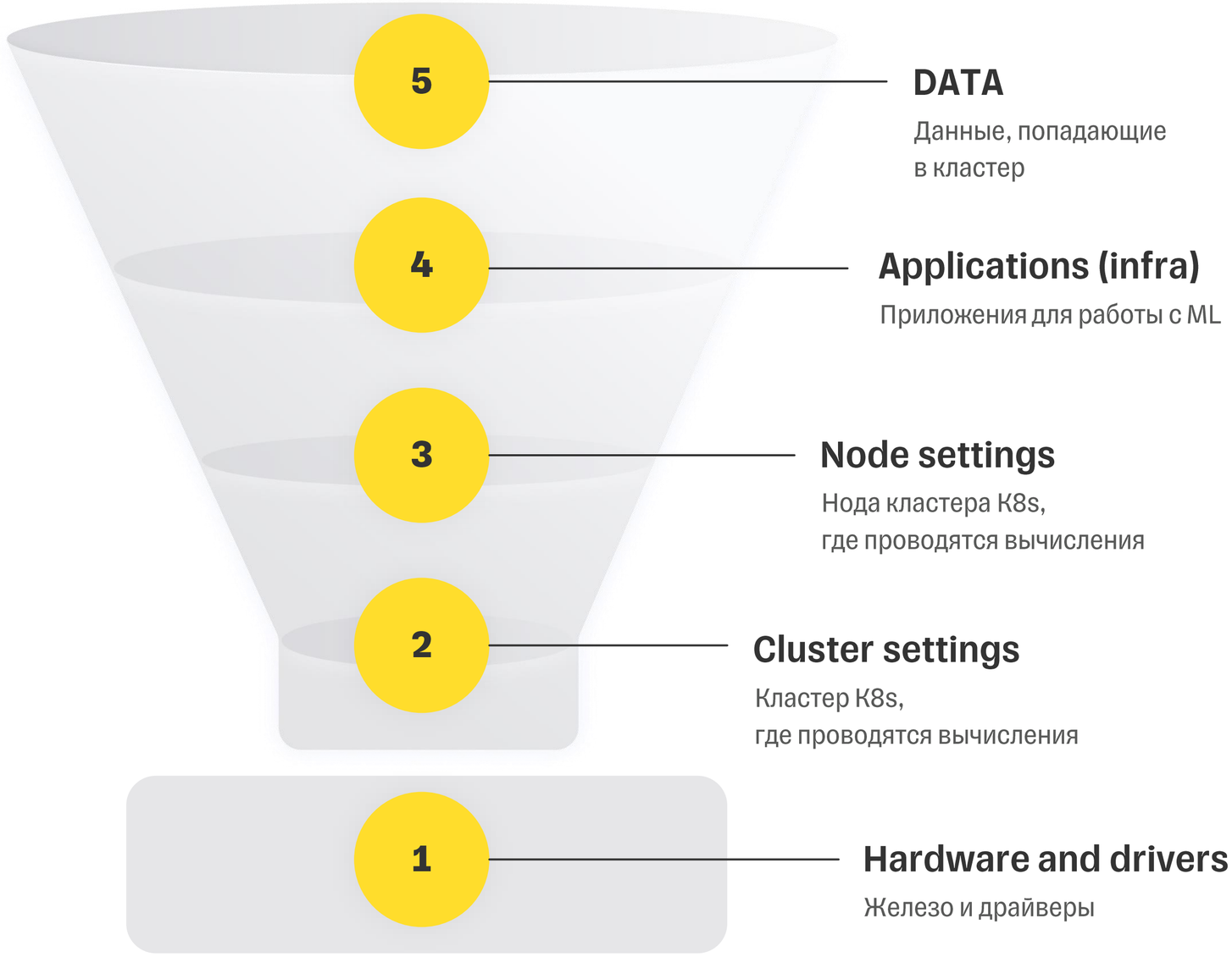


На слайде представлена информация из закрытого исследования, проведенного Т-Банк совместно с компанией ООО “КлаудРан”

Applications (infra)

- Проверить текущие версии приложений, обеспечивающие эксплуатацию GPU нагрузок (например: KubeFlow, Apache Airflow, MLFlow, ClearML)
- Проверить включенность логирования запуска сценариев
- Проверить использование секретов, размещенных в доверенном Vault
- Проверить Service Account используемого компонента executor и убедиться, что он не использует излишних прав
- Проверить сетевые ограничения используемого компонента executor на предмет излишних сетевых доступов
- Проверить разделение служебных компонентов ML и пользовательских нагрузок

Уровни безопасности ML K8s

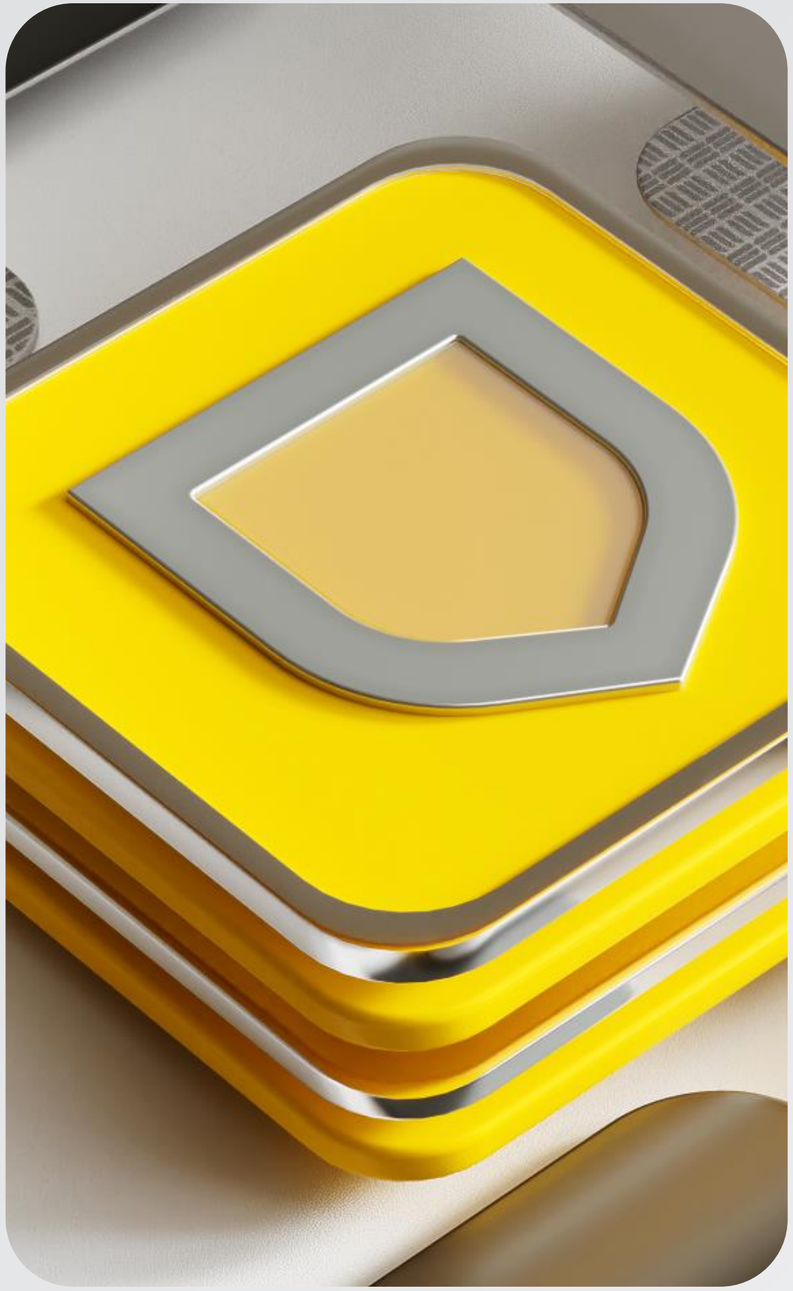


На слайде представлена информация из закрытого исследования, проведенного Т-Банк совместно с компанией ООО “КлаудРан”

DATA

- Проверить получение датасетов только из доверенных источников:
  - расширения файлов
  - пути и команды в Docker-образах/контейнерах
  - контроль Dockerfile
  - контроль размера слоев
- Проверить правила защиты от Data drift и атак типа poisoning
- Проверить защиту от коллизий (перезаписи .csv или версий без контроля)
- Проверить правила совместимости инференса и правил обучения относительно входов
- Проверить ограничение использования датасетов встроенных в образ

Выводы!



Выводы!

- Собирайте K8s кластер под задачу
- Пользуйтесь принципом наименьших привилегий (Security by Design)
- Старайтесь избегать старых технологий, используйте CDI, DRA и обновляйтесь вовремя =)
- Не забывайте контролировать tolerations для всех нагрузочных объектов
- Используйте данный чек-лист для периодических проверок текущего состояния безопасности ML кластеров K8s

Вопросы?

Т-Банк — финансовая онлайн-экосистема, объединяющая полный спектр финансовых услуг для частных лиц и бизнеса

3 июня 2025 Москва, LOFT HALL#2

Конференция по БЕзопасности
КОНтейнеров и контейнерных сред

БЕКОН

LUNTRY

Панченко Николай

Руководитель направления
рантайм защиты | Т-Банк

TG: @yours_rage
Site: www.tbank.ru

LUNTRY Т БАНК