AppSecAutomation:
как внедрить проверки в жизнь разработчиков и не свести их с ума

Алена Жилина

DevSecOps-инженер

Вячеслав Давыдов

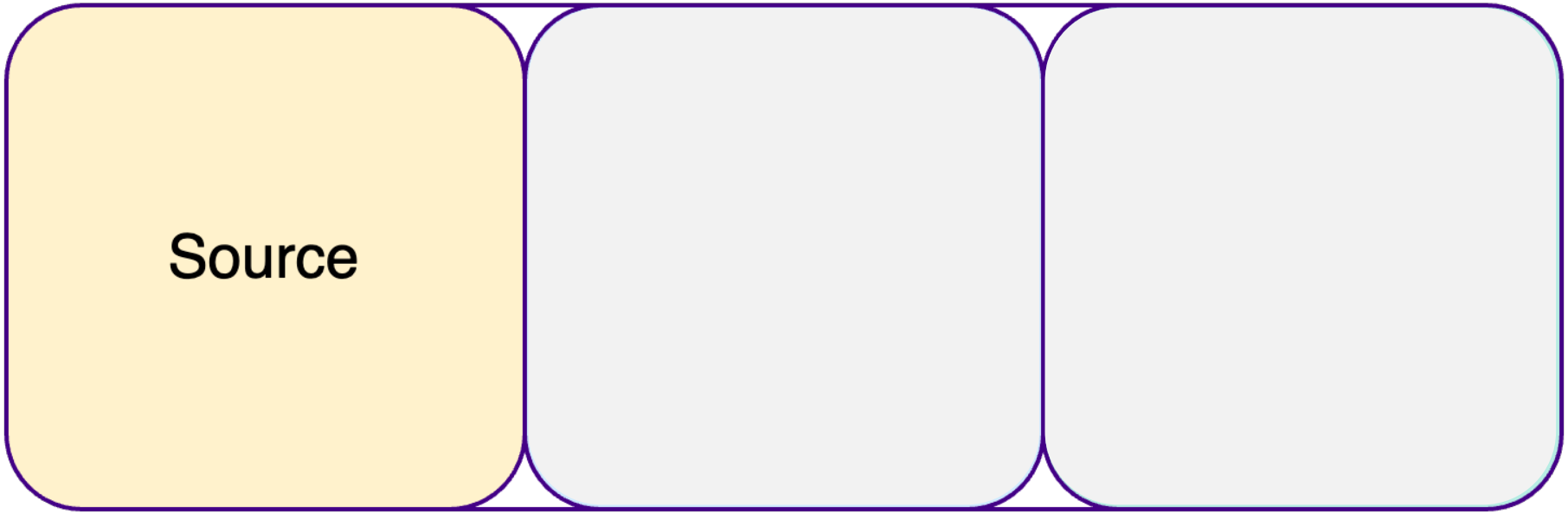
DevSecOps-инженер

О нас

- проектирование и интеграция в AppSec
- безопасность Kubernetes
- data-инжиниринг

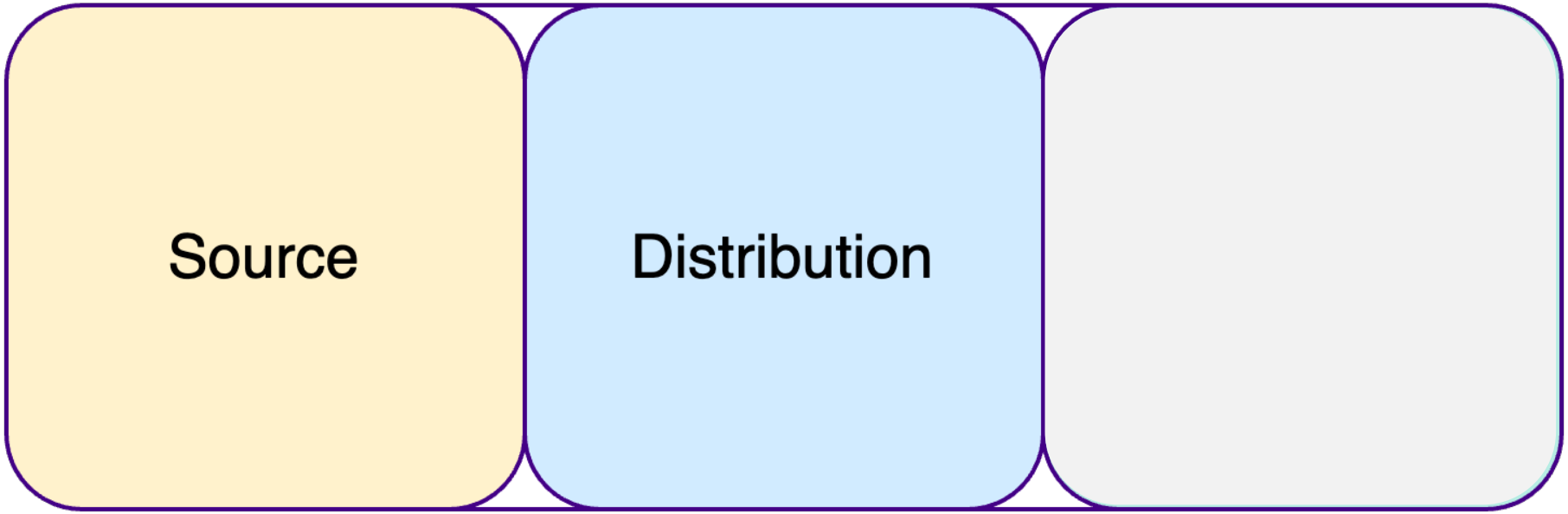
- разработка в AppSec
- CI/CD микросервисов
- геоинформационные системы

С чем работаем



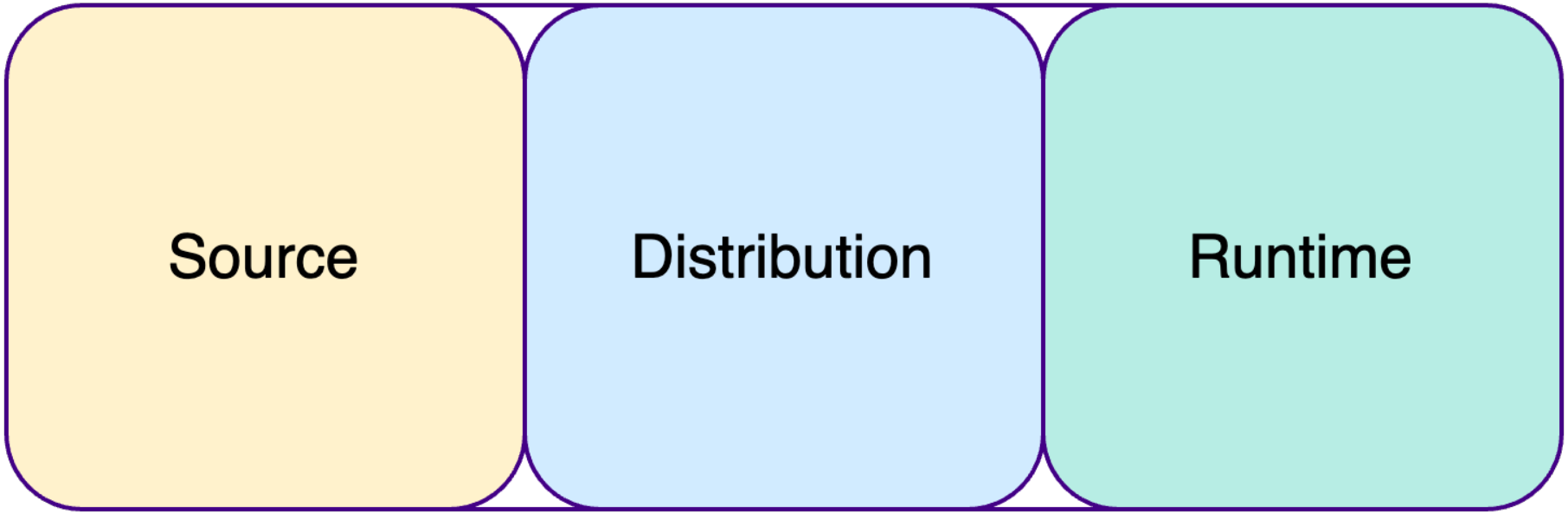
Application

С чем работаем



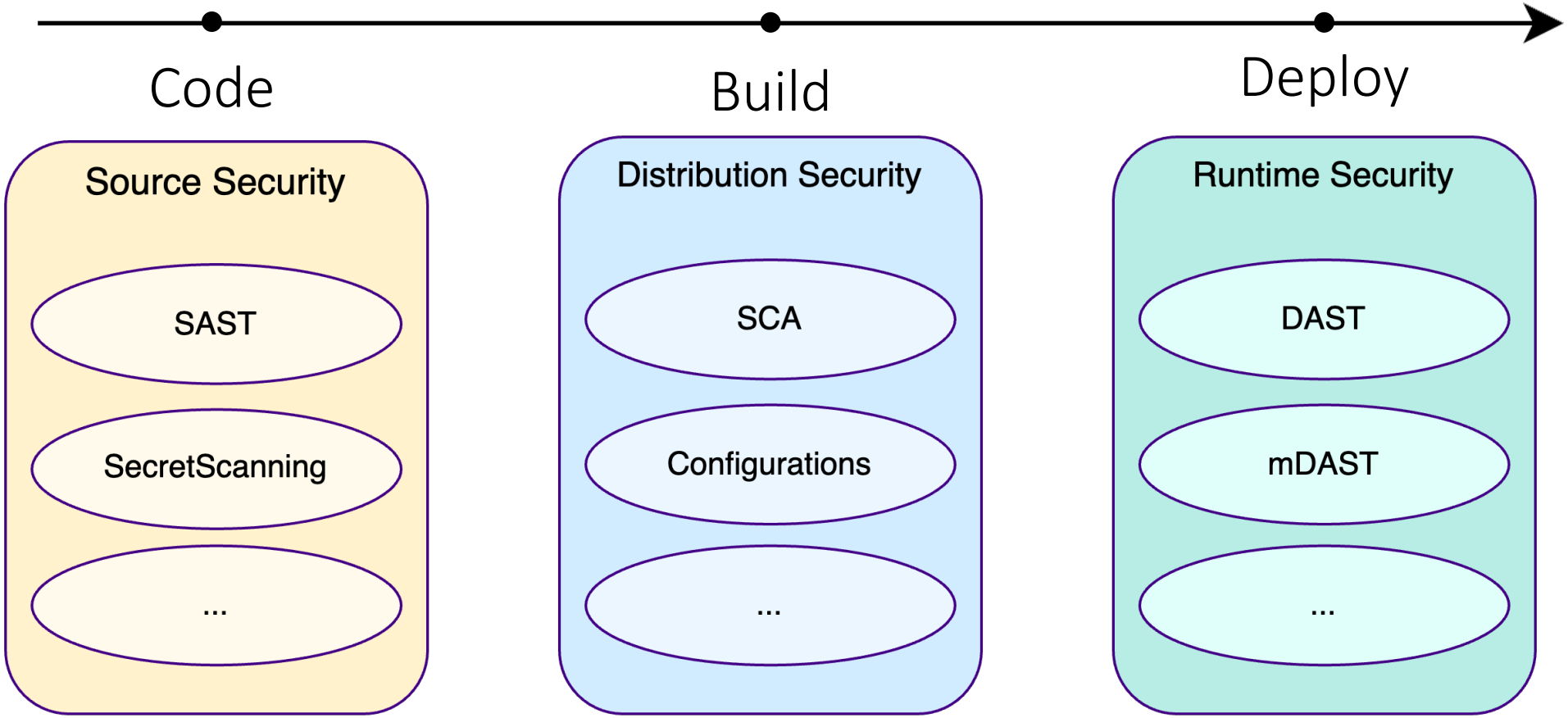
Application

С чем работаем

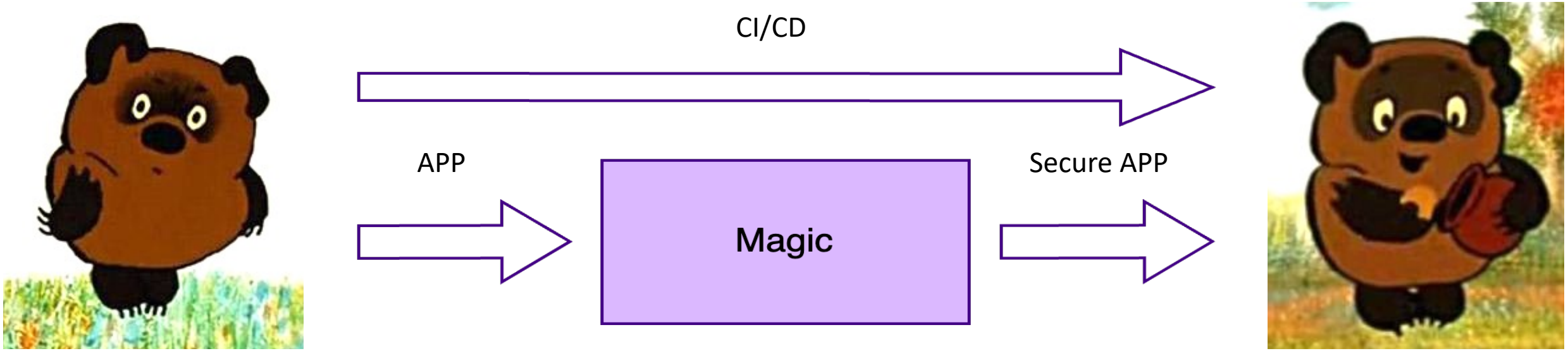


Application

С чем работаем

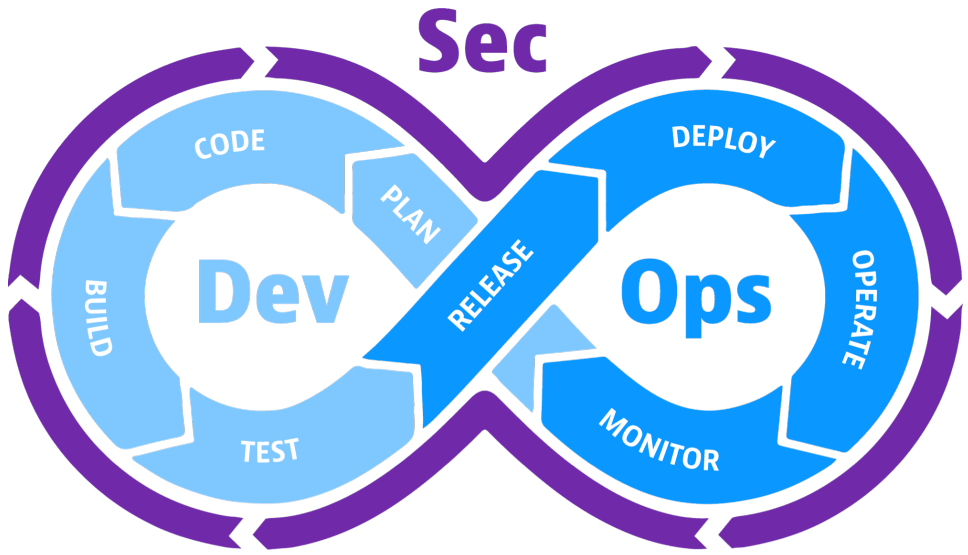


Дивный новый мир



Дивный новый мир

- OnDemand
- Shift left
- триаж и исправление дефектов кода сразу
- перед релизом все баги уже исправлены



Проблематика

команды разработки

- ⇢ сложные абстракции
  - много срочных консультаций перед релизом
  - AppSec-инженерам необходима включённость в DevOps-процессы

SSDLC

SDP

SBOM

OSA

mDAST

SDL

SCA

BCA

SQG

DAST

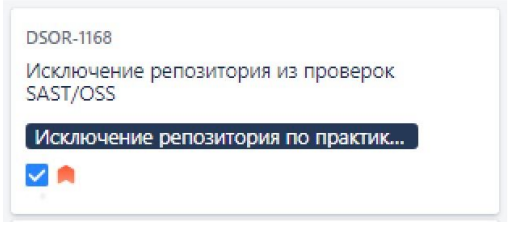
SAST

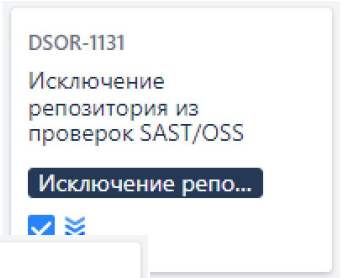
BAST

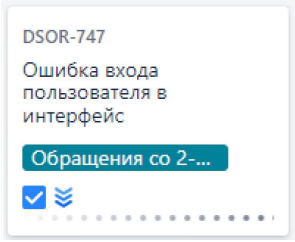
Проблематика

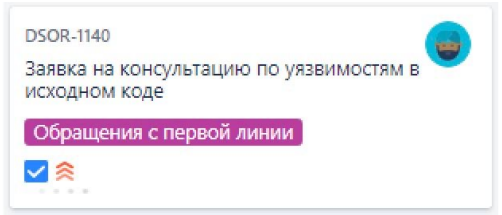
команды разработки

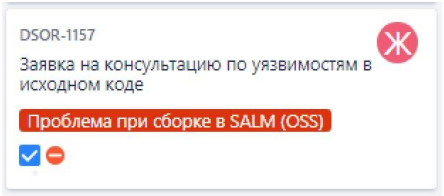
- сложные абстракции
- ⋯➧ много срочных консультаций перед релизом
- AppSec-инженерам необходима включённость в DevOps-процессы









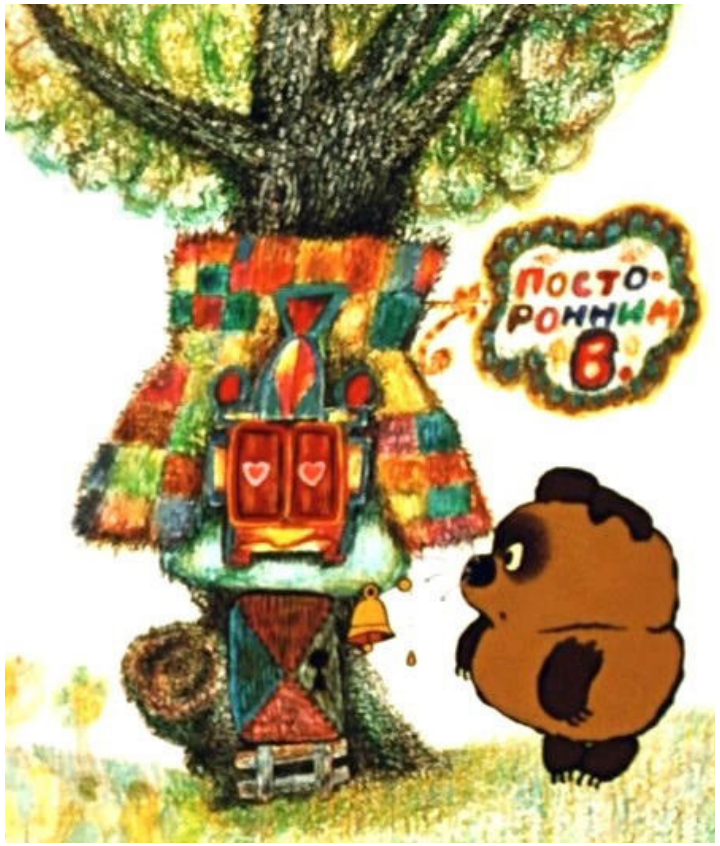


Проблематика

команды разработки

- сложные абстракции
- много срочных консультаций перед релизом

⋯▸ AppSec-инженерам необходима включённость в DevOps-процессы

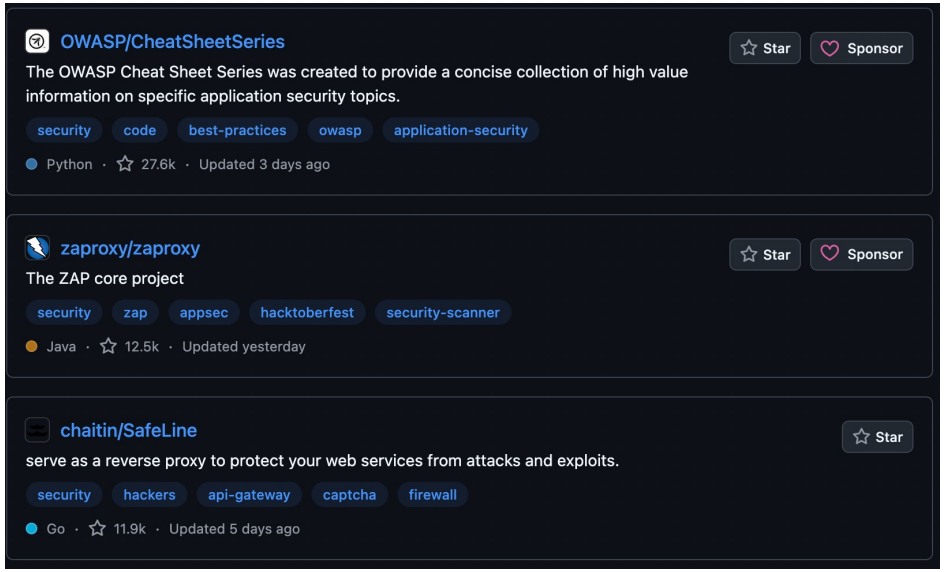


Проблематика

реализация проверок

⋯▸ нужны релевантные инструменты

- автоматизация запуска требует ресурсов
- непонятные отчёты
- onDemand сканирования создают доп нагрузку на ресурсы
- длительные согласования доступов



Проблематика

реализация проверок

- нужны релевантные инструменты
- ⋯▸ автоматизация запуска требует ресурсов
- непонятные отчёты
- onDemand сканирования создают доп нагрузку на ресурсы
- длительные согласования доступов

время на обучение

мотивация

ОЗУ

сетевые ресурсы

CPU

время на тестирование

место на диске

затраты на обучение

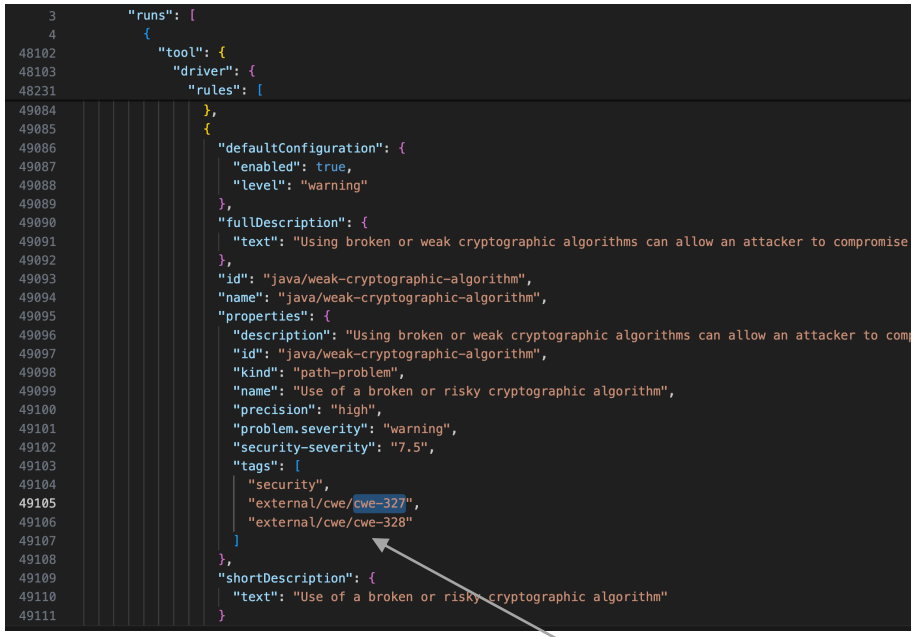
время на интеграцию

затраты на инструменты

Проблематика

реализация проверок

- нужны релевантные инструменты
- автоматизация запуска требует ресурсов
- ⋯▸ непонятные отчёты
- onDemand сканирования создают доп нагрузку на ресурсы
- длительные согласования доступов



CWE-327

Проблематика

реализация проверок

- нужны релевантные инструменты
- автоматизация запуска требует ресурсов
- непонятные отчёты
- ⋯➧ onDemand сканирования создают доп нагрузку на ресурсы
- длительные согласования доступов

Проблематика

реализация проверок

- нужны релевантные инструменты
- автоматизация запуска требует ресурсов
- непонятные отчёты
- onDemand сканирования создают доп нагрузку на ресурсы

⋯▸ длительные согласования доступов

О чём расскажем?

1

Сканирование как ресурс

Управление сканированиями через YAML-интерфейсы

2

Что нужно знать о приложении?

Анализ характеристик артефакта

3

Запуск инструментов

Сценарии запуска сканеров для оптимизации процесса

4

Что дальше?

Работа с отчётом и SecurityQualityGate

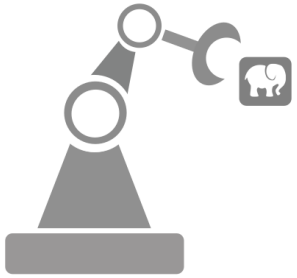
Абстракции:
Kubernetes всюду

Kubernetes-оператор

Software SRE

Программа с навыками опытного администратора

Контроллер, который автоматизирует управление сложными приложениями на кластере, заменяя ручные действия по развертыванию и поддержке.



yaml Copy code

```yaml
apiVersion: postgres-operator.crunchydata.com/v1beta1
kind: PostgresCluster
metadata:
  name: hippo
spec:
  image: registry.developers.crunchydata.com/crunchydata/crunchy-postgres:ubi8-13.4-
0
  postgresVersion: 13
  instances:
    - name: instance1
      replicas: 3
      dataVolumeClaimSpec:
        accessModes:
        - "ReadWriteOnce"
        resources:
          requests:
            storage: 1Gi
  backups:
    pgbackrest:
      image: registry.developers.crunchydata.com/crunchydata/crunchy-pgbackrest:ubi8
-2.33-0
      repos:
      - name: repo1
        volume:
          volumeClaimSpec:
            accessModes:
            - "ReadWriteOnce"
            resources:
              requests:
                storage: 1Gi
```

AppSecTool custom resource

Минимально и достаточно:

- **что может сканировать**
- где расположен

Кастомизация поведения:

- необходимые ресурсы
- набор правил
- типизация артефакта

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecTool
metadata:
  name: semgrep
spec:
  name: semgrep
  toolConfiguration:
  - state: injob
    tag: "mini-python"
    artifactType: "code"
    languages: ["python"]
    size: 200Mi
    image: "my-registry:5000/semgrep:0.1"
    rulesPath:
    - "/etc/semgrep/python"
    cpuRequest: 100m
    memoryRequest: 200Mi

  - state: standAlone
    tag: "custom-secrets"
    artifactType: "code"
    languages: ["java", "python", "yaml"]
    size: 200Mi
    image: "my-registry:5000/semgrep:agent"
    rulesPath:
    - "/etc/semgrep/test/rules"
    host:
      url: "localhost"
      port: 8080
```

AppSecTool custom resource

Минимально и достаточно:

- что может сканировать
- **где расположен**

Кастомизация поведения:

- необходимые ресурсы
- набор правил
- типизация артефакта

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecTool
metadata:
  name: semgrep
spec:
  name: semgrep
  toolConfiguration:
  - state: injob
    tag: "mini-python"
    artifactType: "code"
    languages: ["python"]
    size: 200Mi
    image: "my-registry:5000/semgrep:0.1"
    rulesPath:
    - "/etc/semgrep/python"
    cpuRequest: 100m
    memoryRequest: 200Mi

  - state: standAlone
    tag: "custom-secrets"
    artifactType: "code"
    languages: ["java", "python", "yaml"]
    size: 200Mi
    image: "my-registry:5000/semgrep:agent"
    rulesPath:
    - "/etc/semgrep/test/rules"
    host:
      url: "localhost"
      port: 8080
```

AppSecTool custom resource

Минимально и достаточно:

- что может сканировать
- где расположен

Кастомизация поведения:

- **необходимые ресурсы**
- набор правил
- типизация артефакта

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecTool
metadata:
  name: semgrep
spec:
  name: semgrep
  toolConfiguration:
  - state: injob
    tag: "mini-python"
    artifactType: "code"
    languages: ["python"]
    size: 200Mi
    image: "my-registry:5000/semgrep:0.1"
    rulesPath:
    - "/etc/semgrep/python"
    cpuRequest: 100m
    memoryRequest: 200Mi

  - state: standAlone
    tag: "custom-secrets"
    artifactType: "code"
    languages: ["java", "python", "yaml"]
    size: 200Mi
    image: "my-registry:5000/semgrep:agent"
    rulesPath:
    - "/etc/semgrep/test/rules"
    host:
      url: "localhost"
      port: 8080
```

AppSecTool custom resource

Минимально и достаточно:

- что может сканировать
- где расположен

Кастомизация поведения:

- необходимые ресурсы
- **набор правил**
- типизация артефакта

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecTool
metadata:
  name: semgrep
spec:
  name: semgrep
  toolConfiguration:
  - state: injob
    tag: "mini-python"
    artifactType: "code"
    languages: ["python"]
    size: 200Mi
    image: "my-registry:5000/semgrep:0.1"
    rulesPath:
    - "/etc/semgrep/python"
    cpuRequest: 100m
    memoryRequest: 200Mi

  - state: standAlone
    tag: "custom-secrets"
    artifactType: "code"
    languages: ["java", "python", "yaml"]
    size: 200Mi
    image: "my-registry:5000/semgrep:agent"
    rulesPath:
    - "/etc/semgrep/test/rules"
    host:
      url: "localhost"
      port: 8080
```

AppSecTool custom resource

Минимально и достаточно:

- что может сканировать
- где расположен

Кастомизация поведения:

- необходимые ресурсы
- набор правил
- ⋯› **типизация артефакта**

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecTool
metadata:
  name: semgrep
spec:
  name: semgrep
  toolConfiguration:
  - state: injob
    tag: "mini-python"
    artifactType: "code"
    languages: ["python"]
    size: 200Mi
    image: "my-registry:5000/semgrep:0.1"
    rulesPath:
    - "/etc/semgrep/python"
    cpuRequest: 100m
    memoryRequest: 200Mi

  - state: standAlone
    tag: "custom-secrets"
    artifactType: "code"
    languages: ["java", "python", "yaml"]
    size: 200Mi
    image: "my-registry:5000/semgrep:agent"
    rulesPath:
    - "/etc/semgrep/test/rules"
    host:
      url: "localhost"
      port: 8080
```

AppSecJob custom resource

Минимально и достаточно:

- **объект сканирования**

Кастомизация поведения:

- доступ к объекту сканирования
- менеджер секретов
- представление результата

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecJob
metadata:
  name: appsecjob-3
spec:
  artifact:
    code:
      url: ssh://git@bb.mycompany.ru:project/repo.git
      commithash: c8410457c23b57cf04910548d099c28bb39f3c3f
      source:
        secretName: bitbucket
        secretKeys:
          - sshkey
    distribution:
      hash: 541bc8e0cf501c53f8eeae08c3fc8b22b4212cdc
  secretManager:
    type: vault
    namespace: APPSEC
      path: ddd
    host:
      name: secman.mycompany.ru
      port: 443
  resultDesination:
    type: s3
    host:
      name: minio.mycompany.ru
      port: 443
      bucket: appsec-results
```

AppSecJob custom resource

Минимально и достаточно:

- объект сканирования

Кастомизация поведения:

- **доступ к объекту сканирования**
- менеджер секретов
- представление результата

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecJob
metadata:
  name: appsecjob-3
spec:
  artifact:
    code:
      url: ssh://git@bb.mycompany.ru:project/repo.git
      commithash: c8410457c23b57cf04910548d099c28bb39f3c3f
      source:
        secretName: bitbucket
        secretKeys:
          - sshkey
    distribution:
      hash: 541bc8e0cf501c53f8eeae08c3fc8b22b4212cdc
  secretManager:
    type: vault
    namespace: APPSEC
      path: ddd
    host:
      name: secman.mycompany.ru
      port: 443
  resultDesination:
    type: s3
    host:
      name: minio.mycompany.ru
      port: 443
      bucket: appsec-results
```

AppSecJob custom resource

Минимально и достаточно:

- объект сканирования

Кастомизация поведения:

- доступ к объекту сканирования
- **менеджер секретов**
- представление результата

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecJob
metadata:
  name: appsecjob-3
spec:
  artifact:
    code:
      url: ssh://git@bb.mycompany.ru:project/repo.git
      commithash: c8410457c23b57cf04910548d099c28bb39f3c3f
      source:
        secretName: bitbucket
        secretKeys:
          - sshkey
    distribution:
      hash: 541bc8e0cf501c53f8eeae08c3fc8b22b4212cdc
  secretManager:
    type: vault
    namespace: APPSEC
      path: ddd
    host:
      name: secman.mycompany.ru
      port: 443
  resultDesination:
    type: s3
    host:
      name: minio.mycompany.ru
      port: 443
      bucket: appsec-results
```

AppSecJob custom resource

Минимально и достаточно:

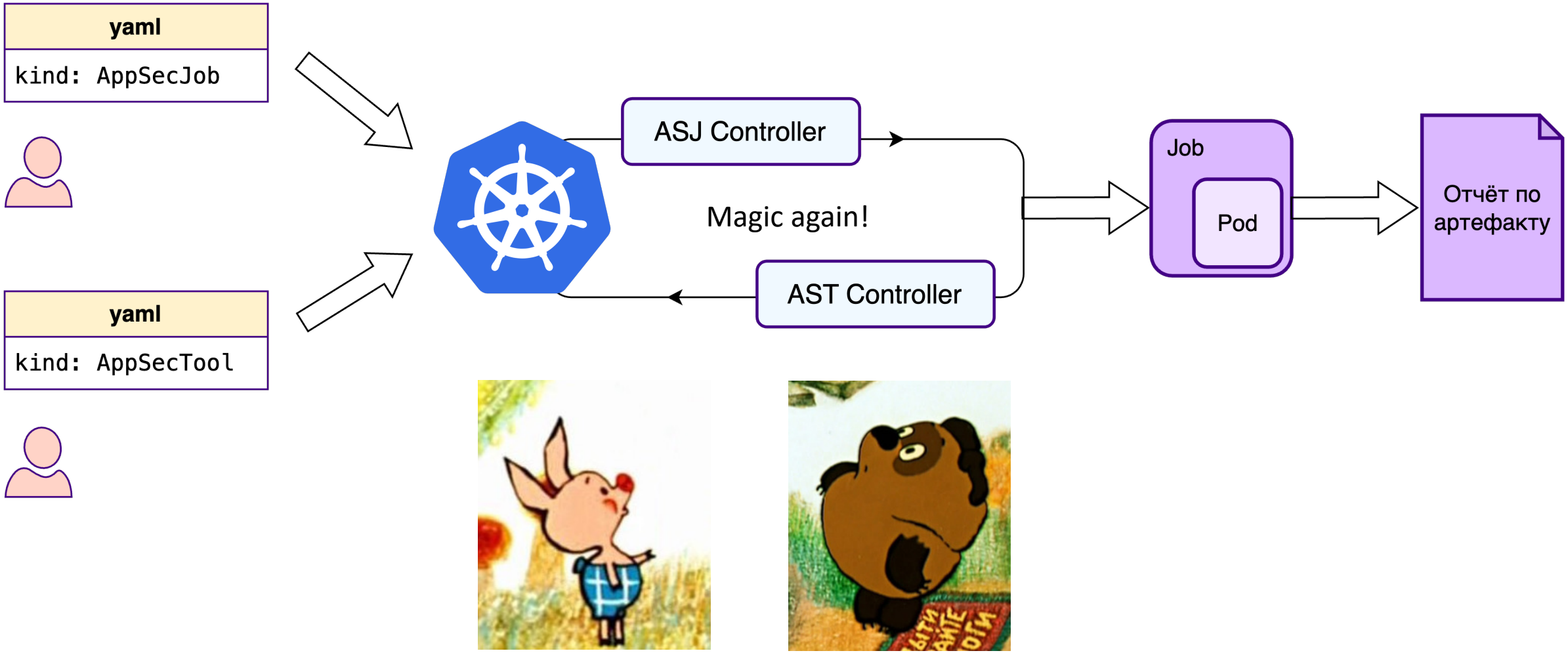
- объект сканирования

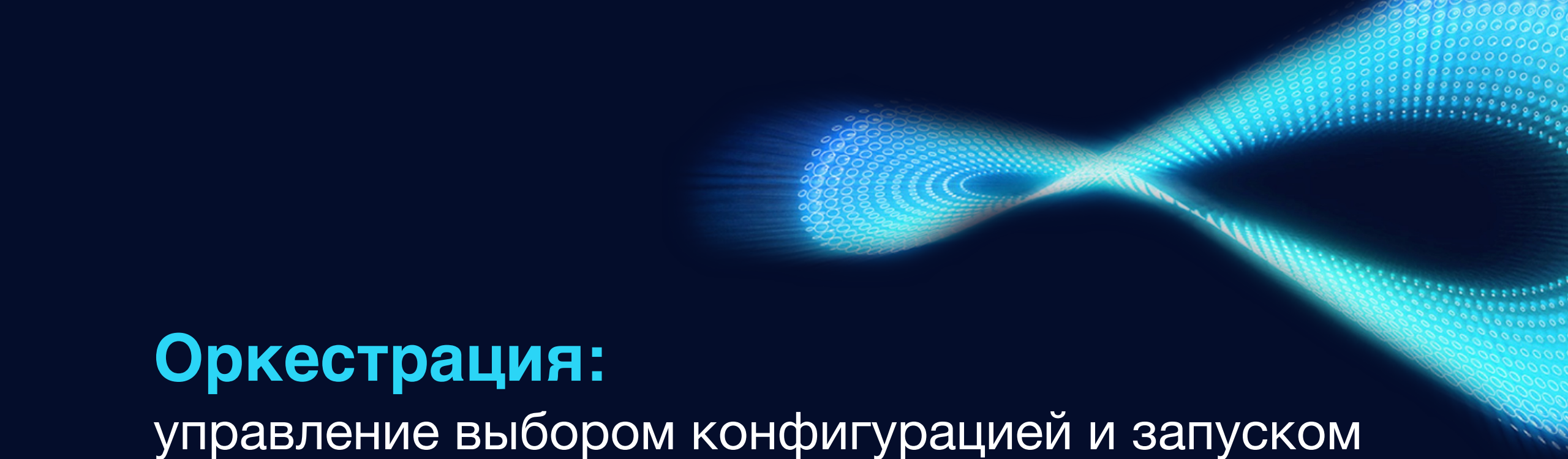
Кастомизация поведения:

- доступ к объекту сканирования
- менеджер секретов
- **представление результата**

```yaml
---
apiVersion: cache.appsec.ru/v1alpha1
kind: AppSecJob
metadata:
  name: appsecjob-3
spec:
  artifact:
    code:
      url: ssh://git@bb.mycompany.ru:project/repo.git
      commithash: c8410457c23b57cf04910548d099c28bb39f3c3f
      source:
        secretName: bitbucket
        secretKeys:
          - sshkey
    distribution:
      hash: 541bc8e0cf501c53f8eeae08c3fc8b22b4212cdc
  secretManager:
    type: vault
    namespace: APPSEC
      path: ddd
    host:
      name: secman.mycompany.ru
      port: 443
  resultDesination:
    type: s3
    host:
      name: minio.mycompany.ru
      port: 443
      bucket: appsec-results
```

AppSecJob custom resource

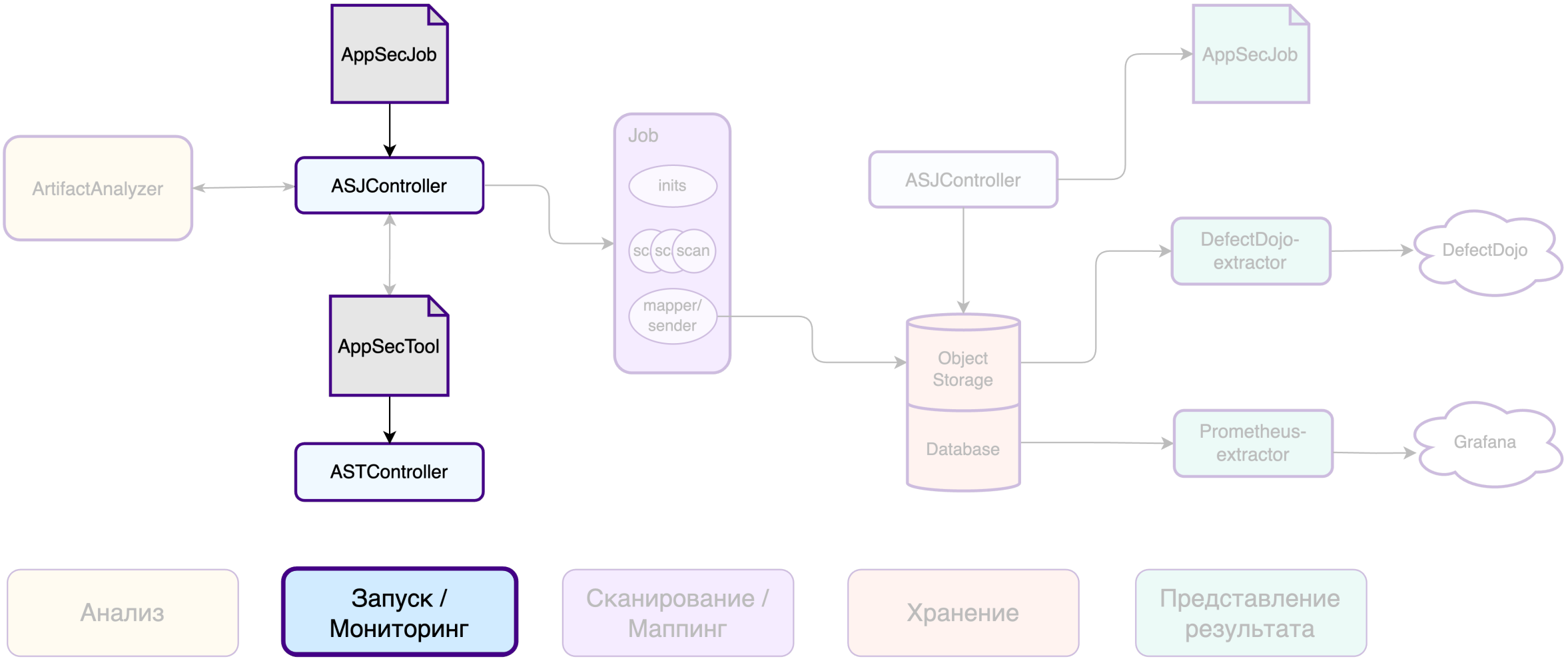




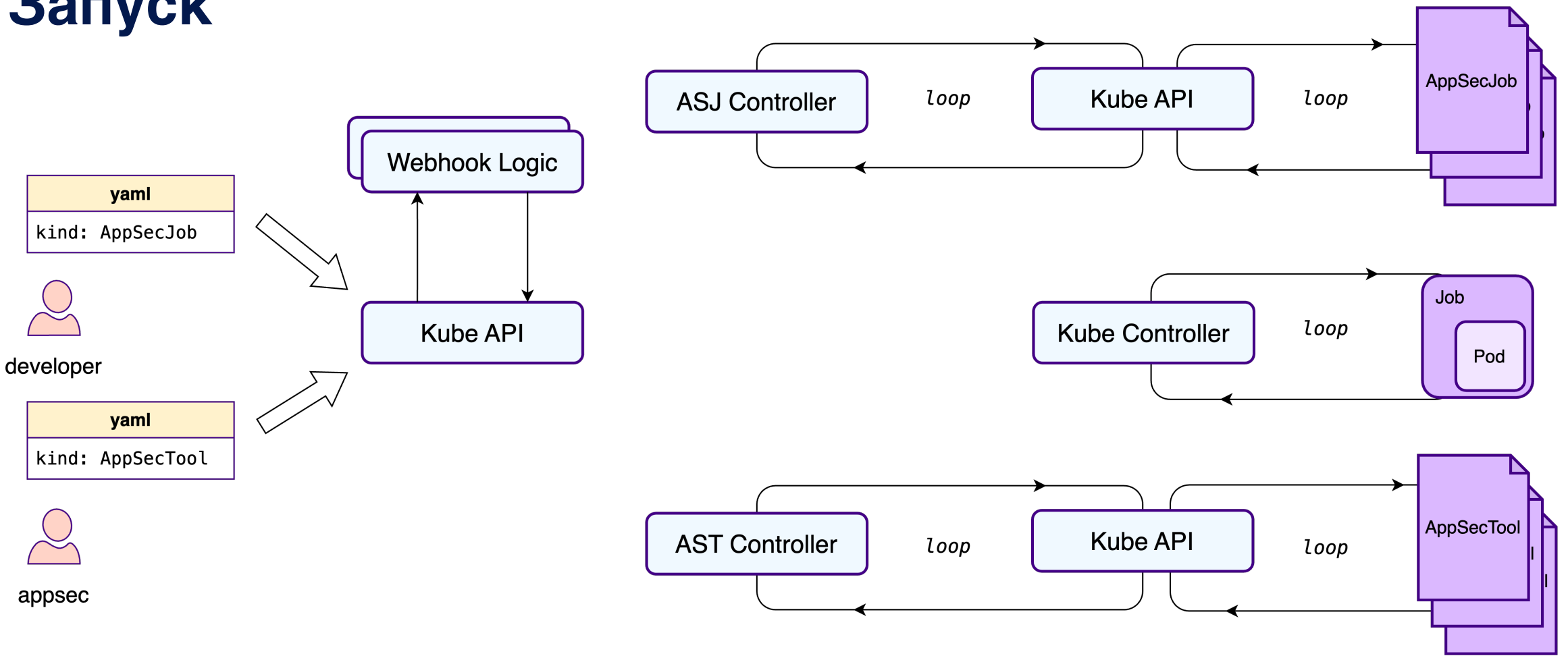
Оркестрация:

управление выбором конфигурацией и запуском AppSec инструментов для получения отчета безопасности

Запуск


AppSecJob
ArtifactAnalyzer
ASJController
AppSecTool
ASTController
Job
inits
sc
sc
scan
mapper/
sender
ASJController
AppSecJob
Object
Storage
Database
DefectDojo-
extractor
DefectDojo
Prometheus-
extractor
Grafana
Анализ
Запуск /
Мониторинг
Сканирование /
Маппинг
Хранение
Представление
результата

Запуск



Клиент

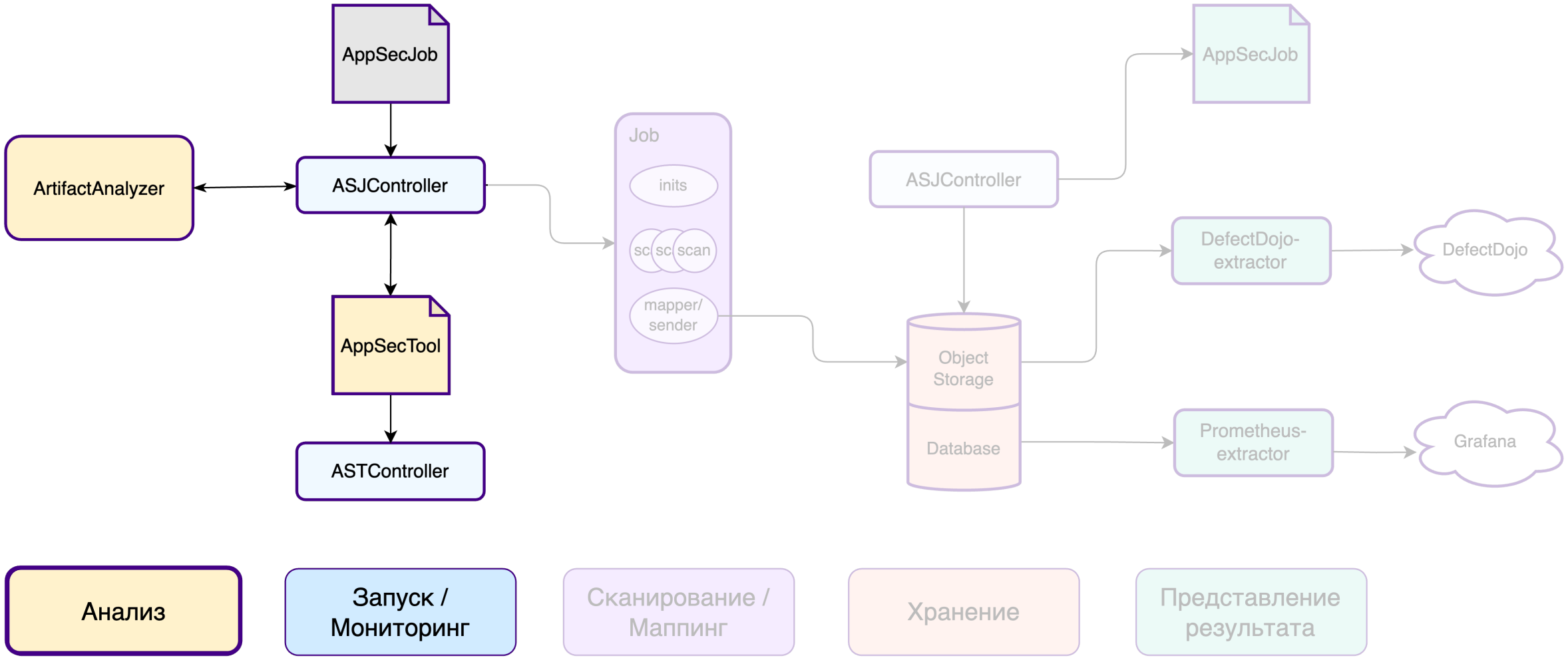
Control Plane

Namespace AppSec

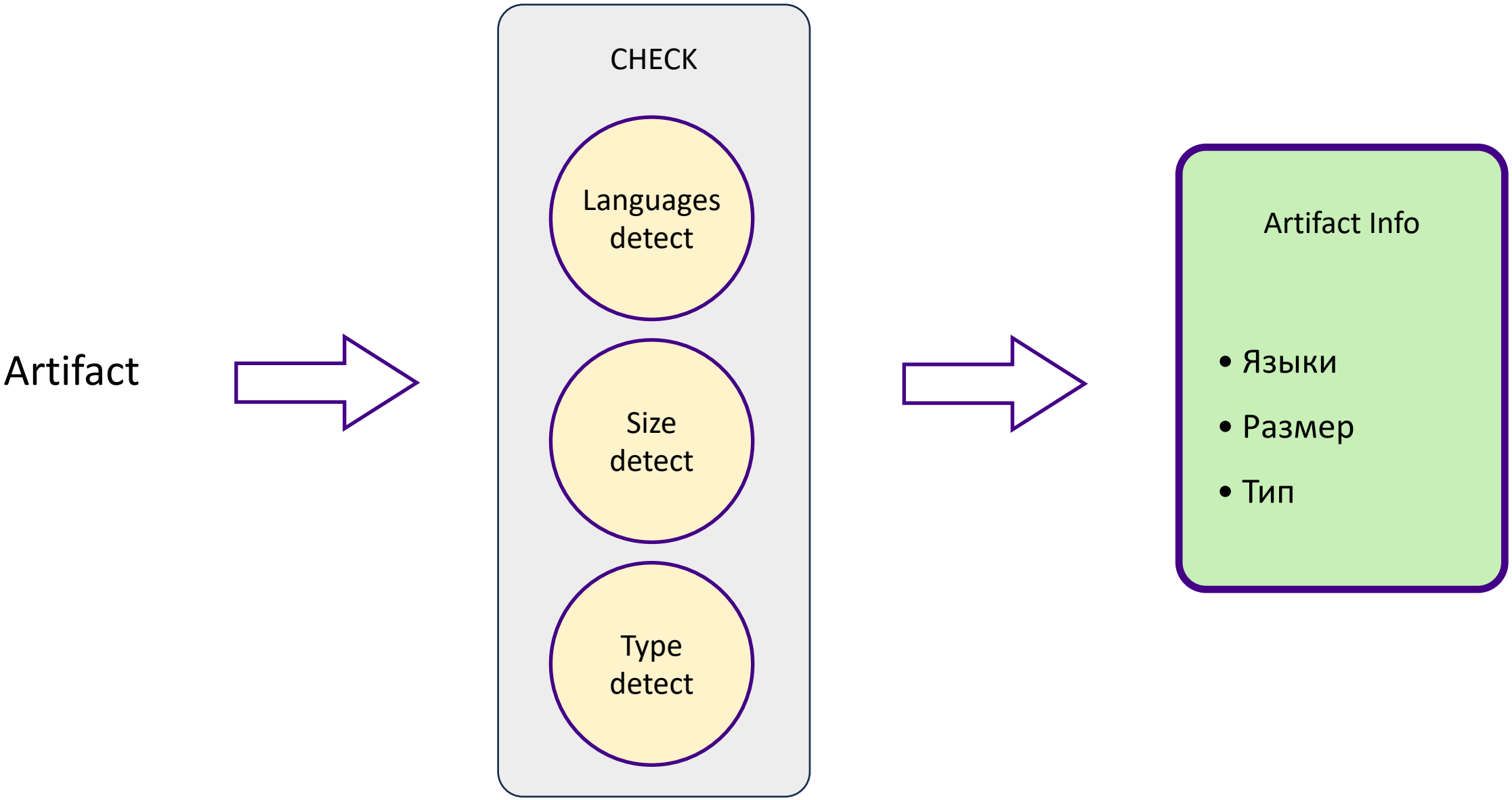
Control Plane

Namespace Клиента

Анализ

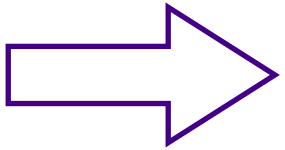

AppSecJob
ArtifactAnalyzer
ASJController
AppSecTool
ASTController
Job
inits
sc
sc
scan
mapper/
sender
ASJController
AppSecJob
DefectDojo-
extractor
DefectDojo
Object
Storage
Database
Prometheus-
extractor
Grafana
Анализ
Запуск /
Мониторинг
Сканирование /
Маппинг
Хранение
Представление
результата

Анализ


Artifact
CHECK
Languages detect
Size detect
Type detect
Artifact Info
• Языки
• Размер
• Тип

Анализ

yaml
`kind: AppSecTool`

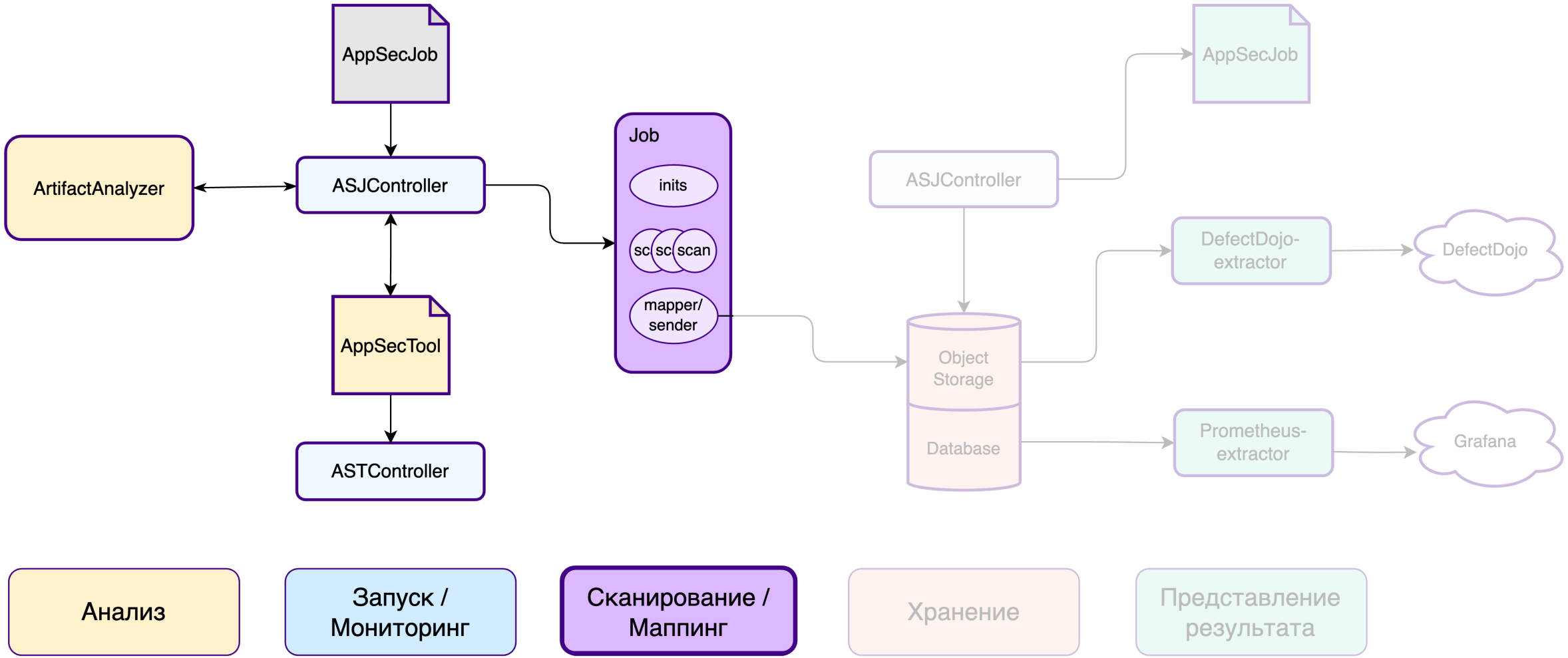


Перечень релевантных инструментов и конфигураций

Artifact Info

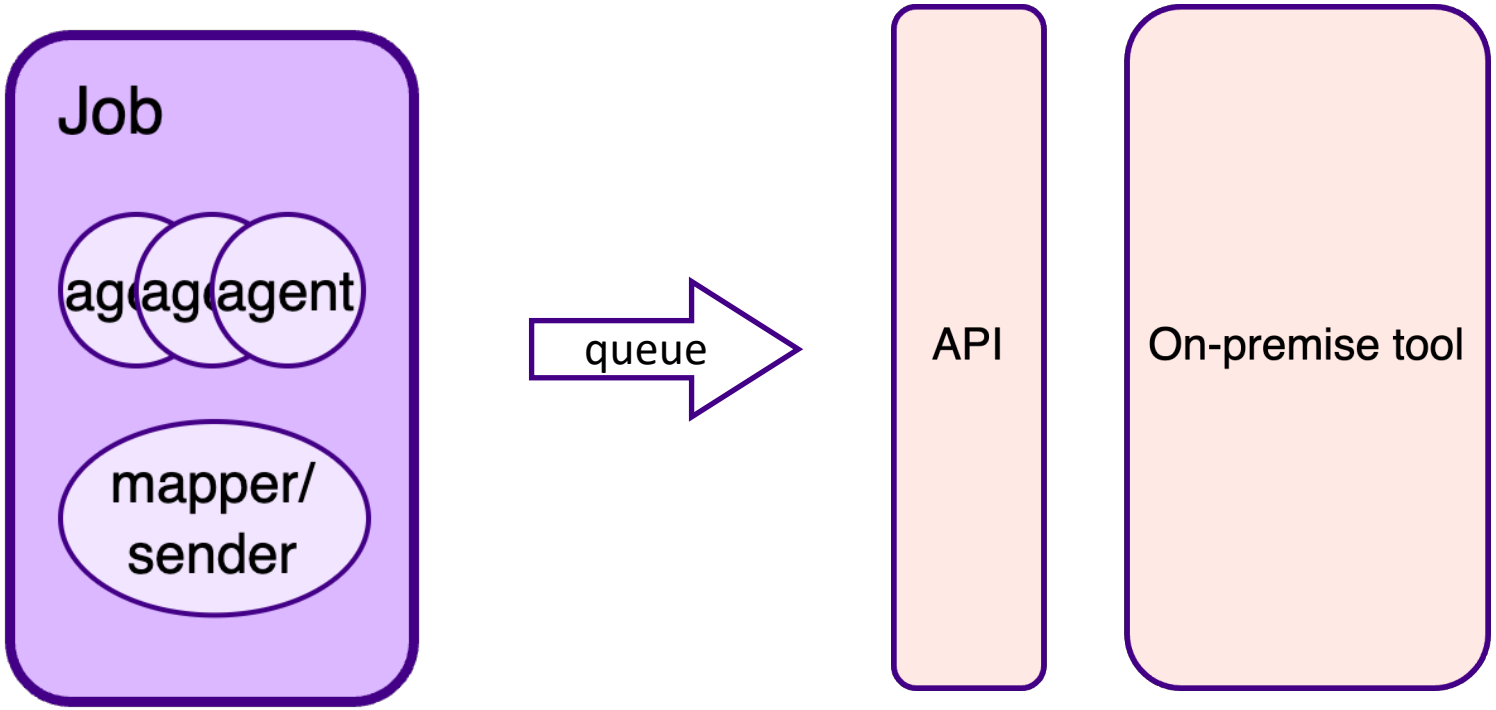
- Языки
- Размер
- Тип

Сканирование



Сканирование

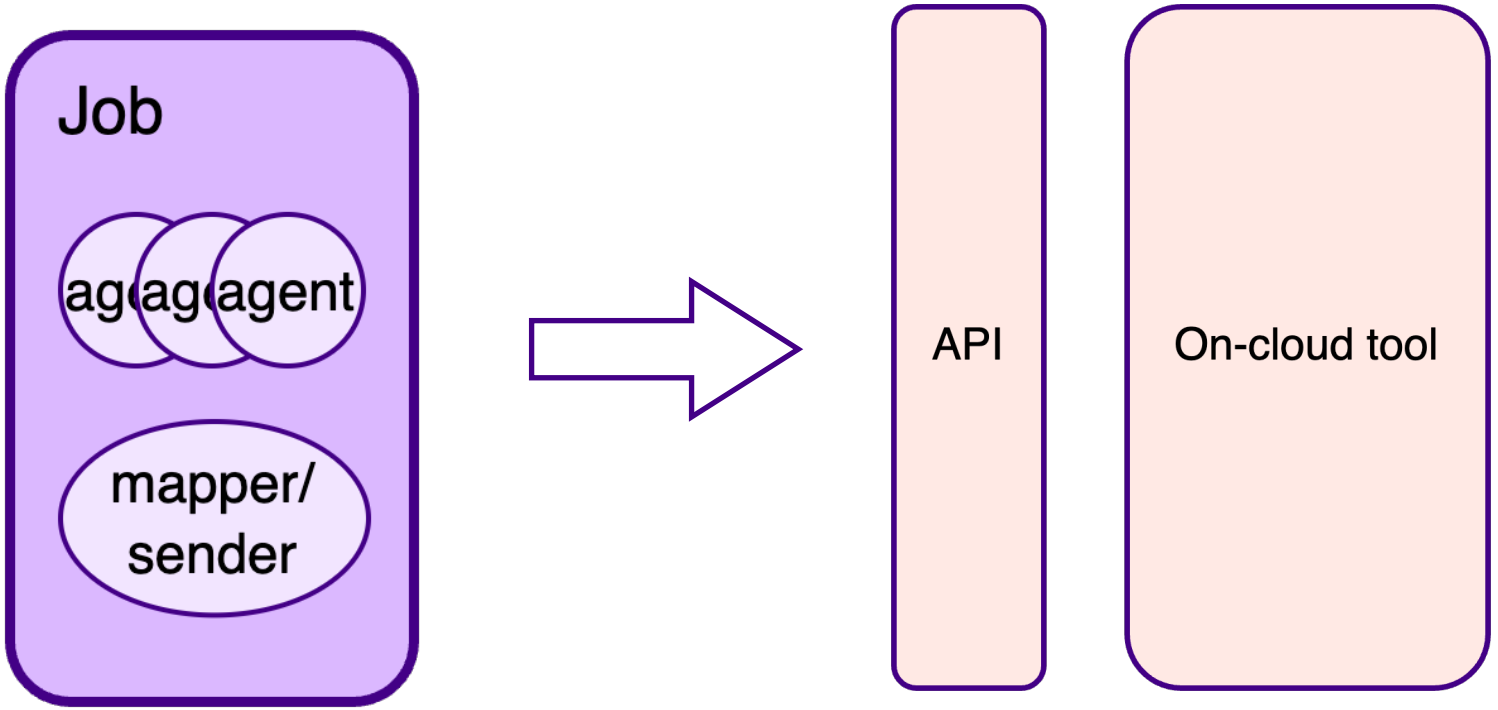
агент к инструменту



- «коробочное» решение
- legacy

Сканирование

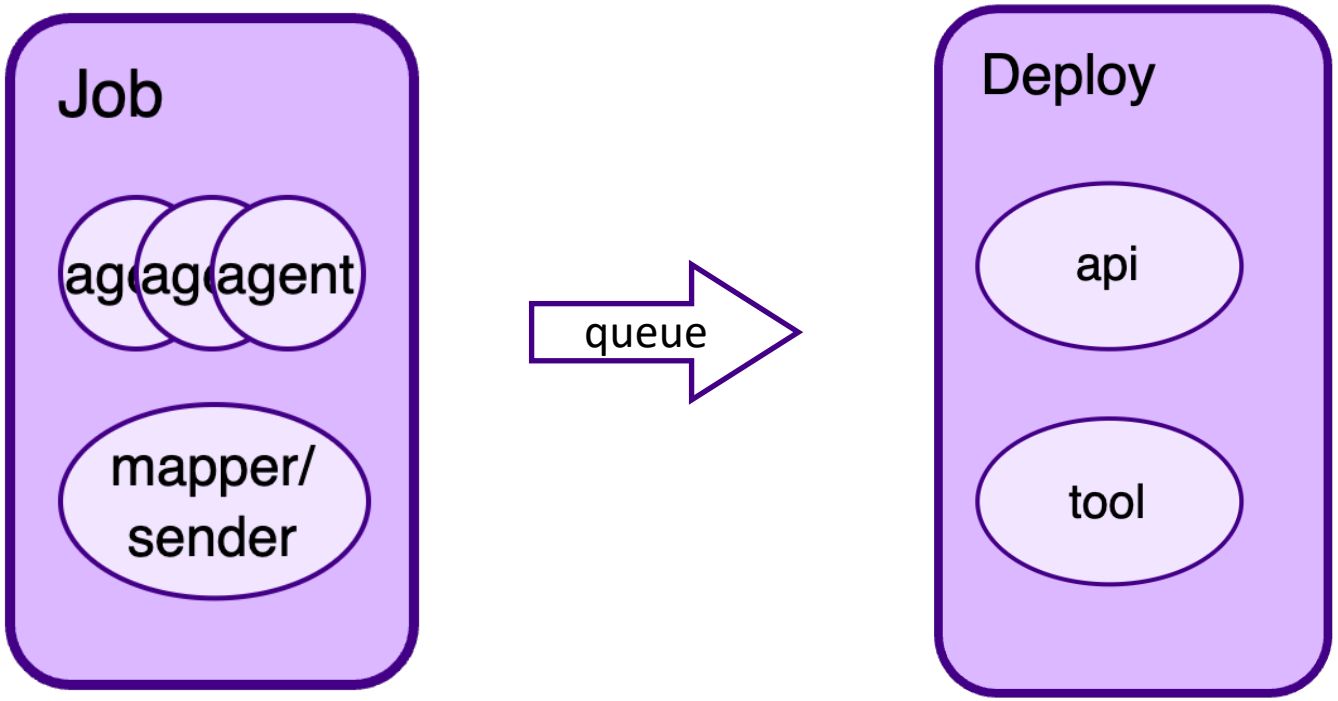
агент к инструменту



- лицензия от вендора

Сканирование

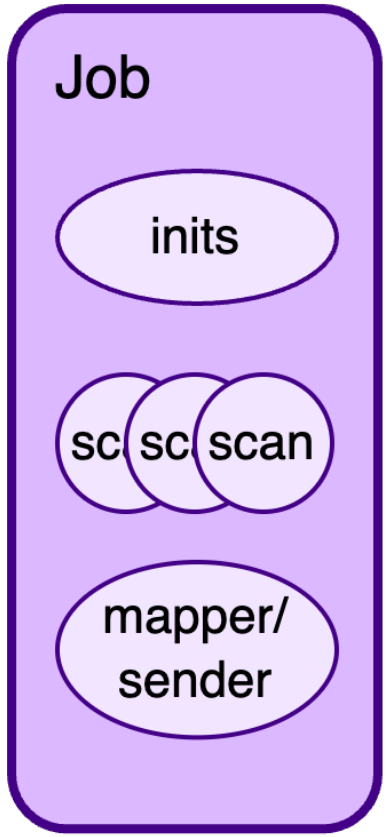
агент к инструменту



- инструмент поддерживает отдельная команда
- реплики по количеству ресурсов

Сканирование

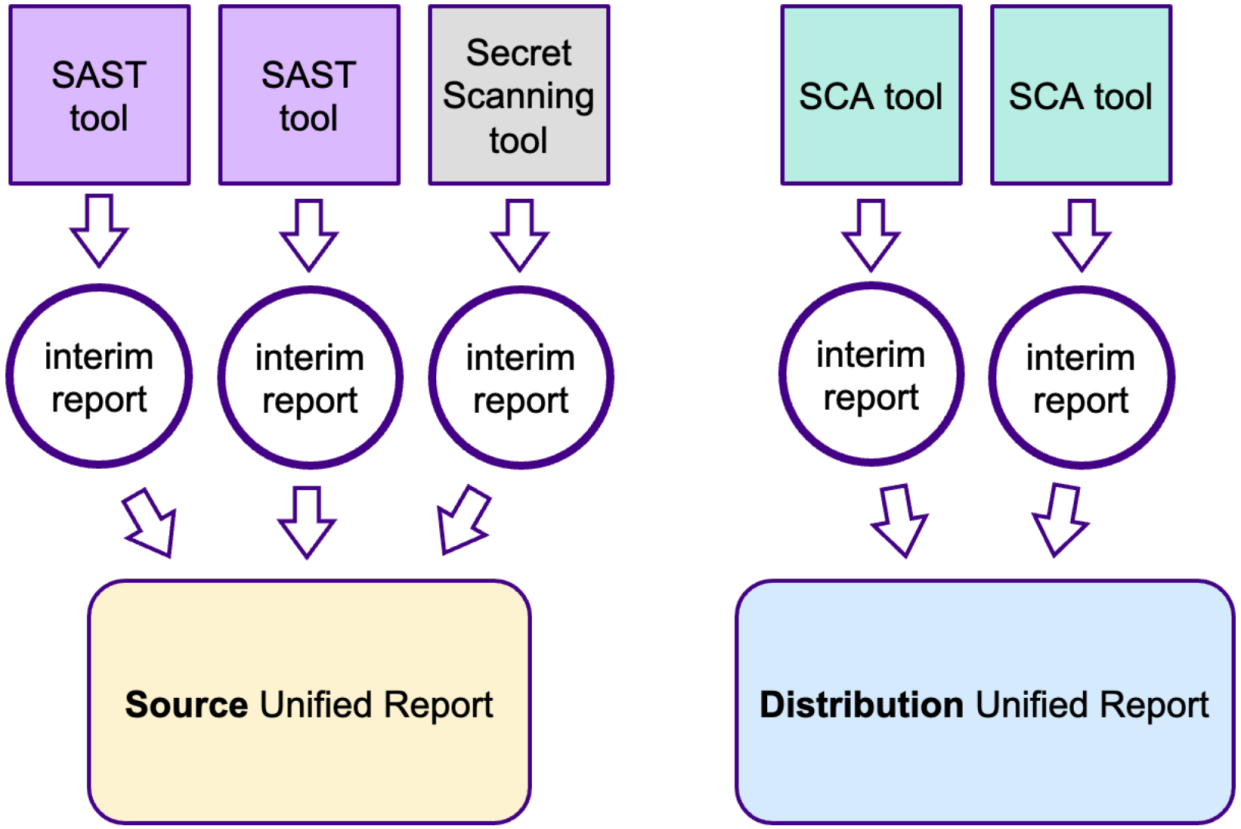
инструмент



- инструмент в контейнере
- получение артефакта в init-контейнере
- выделение необходимого количества ресурсов под конкретный запуск
- нет очередей
- все артефакты сканирования в области работы джобы

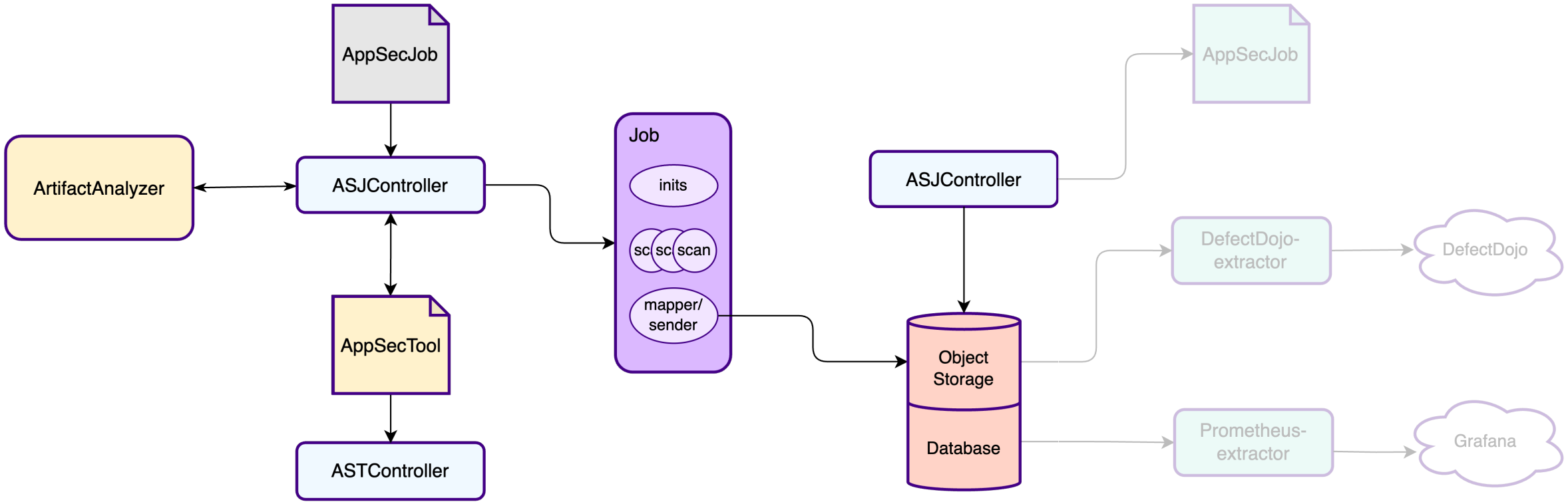
- ? медленнее старт
- ? обновление правил

Маппинг



- Объединяем сработки из нескольких отчетов
- Ищем пересечение
- Повышаем уверенность в их релевантности
- Все сработки приводим в унифицированный формат

Хранение



Анализ

Запуск / Мониторинг

Сканирование / Маппинг

Хранение

Представление результата

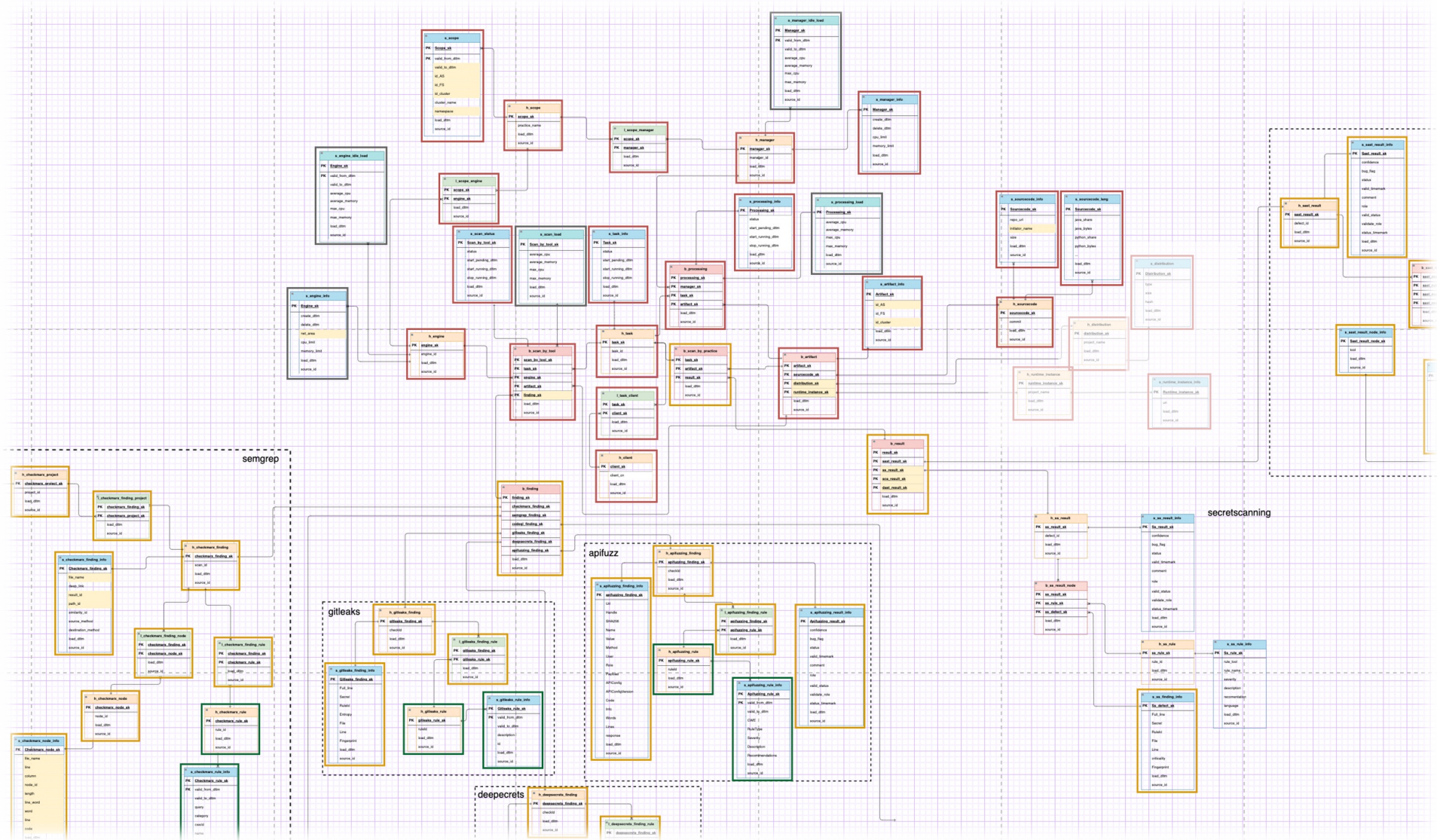
Хранение

- отражение ключевых бизнес-сущностей
- управление дефектами
- интеграция с системой триажа
- отражение данных аудита и мониторинга

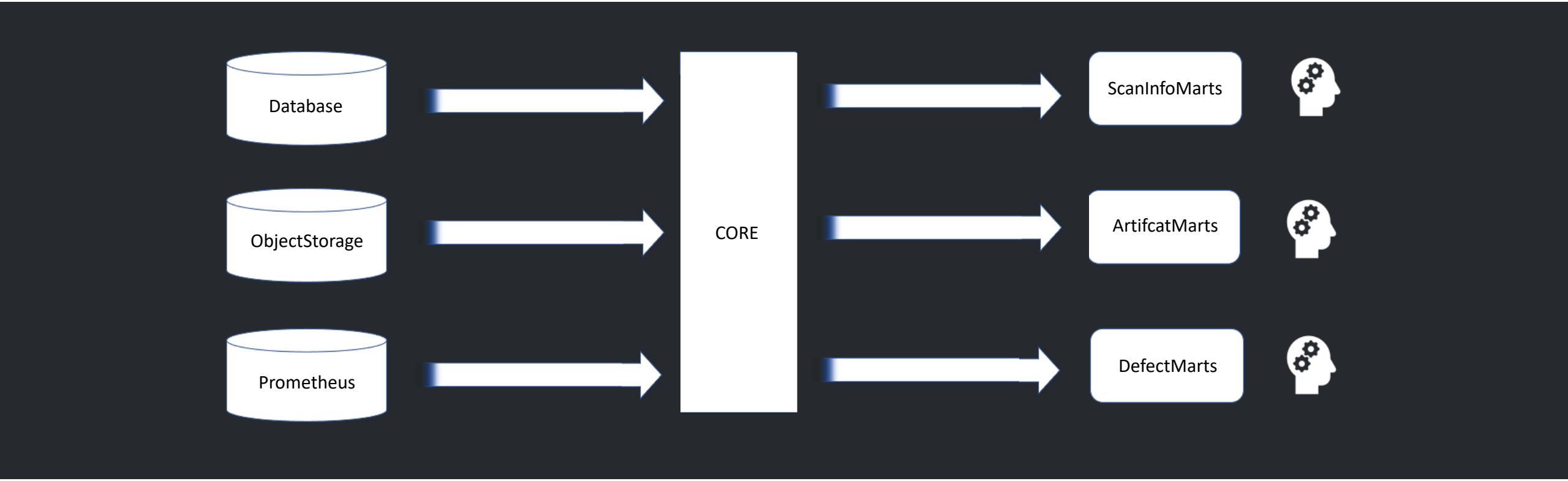
Хранение

DWH

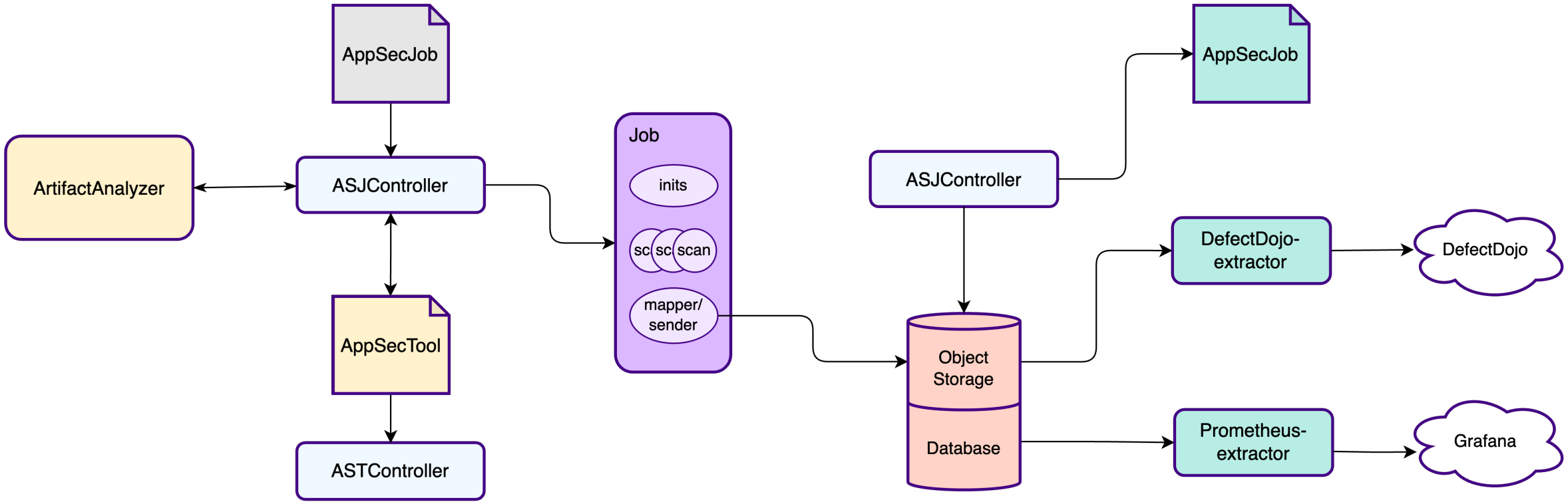
DATA VAULT



Хранение


Database
ObjectStorage
Prometheus
CORE
ScanInfoMarts
ArtifcatMarts
DefectMarts

Представление результата



Анализ

Запуск / Мониторинг

Сканирование / Маппинг

Хранение

Представление результата

Представление результата

- «мета»-результат в статусе AppSecJob
- дефекты в DefectDojo --> триаж --> интеграция с Jira
- метрики работы инструментов в Prometheus --> построение дашбордов в Grafana

NAME	STATUS	LOW	MEDIUM	HIGH	CRITICAL	TOP 3 VULNERABILITIES
appsecjob-1	So-so	18	5	0	0	1. CWE-79-3 (5.4): Cross-site Scripting (XSS) in User Comments 2. CWE-200 (5.0): Exposure of Sensitive Information in Error Messages 3. CWE-20 (3.1): Improper Input Validation in API Endpoints
appsecjob-2	Feel bad	12	3	1	0	1. CVE-2023-12288 (8.2): Remote Code Execution (RCE) via Unsanitized Input 2. CWE-200 (5.0): Exposure of Sensitive Information to an Unauthorized Actor 3. CWE-26 (4.6): Improper Input Validation in File Uploads
appsecjob-3	Critical	22	2	2	1	1. CVE-2023-67890 (9.1): SQL Injection in User Login Form 2. CVE-2023-54449 (8.6): Remote Code Execution (RCE) via Insecure Deserialization 3. CWE-200 (7.9): Exposure of Sensitive Information to an Unauthorized Actor
appsecjob-4	Great	15	2	0	0	1. CWE-20 (2.8): Improper Input Validation in API Endpoints 2. CWE-200 (3.5): Exposure of Sensitive Information in Error Messages 3. CWE-20 (2.5): Improper Input Validation in File Uploads

Security Quality Gate


```
      attestations:
        - predicateType: "cosign.sigstore.dev/attestation/vuln/v1"
          conditions:
            - all:
              - key: "{{ images[0].attestations[?(@.predicateType ==
          'cosign.sigstore.dev/attestation/vuln/v1')].details.vulnerabilities[?(@.severity ==
          'Critical')] | length(@) }}"
                operator: GreaterThan
                value: 0
            - all:
              - key: "{{ images[0].attestations[?(@.predicateType ==
          'cosign.sigstore.dev/attestation/vuln/v1')].details.vulnerabilities[?(@.severity ==
          'High')] | length(@) }}"
                operator: GreaterThan
                value: 2
  preconditions:
    all:
      - key: "{{ images[0].attestations[?(@.predicateType ==
  'cosign.sigstore.dev/attestation/vuln/v1')].predicateType }}"
        operator: Equals
        value: "cosign.sigstore.dev/attestation/vuln/v1"
```

```
validate:
    message: |
      Образ не прошел проверку из-за следующих уязвимостей:
      - Критические: {{ images[0].attestations[?(@.predicateType == 'cosign.sigstore.dev/attestation/vuln/v1')].details.vulnerabilities[?(@.severity ==
      'Critical')] | length(@) }}
      - Высокие: {{ images[0].attestations[?(@.predicateType == 'cosign.sigstore.dev/attestation/vuln/v1')].details.vulnerabilities[?(@.severity == 'High')]
      | length(@) }}
      - Средние: {{ images[0].attestations[?(@.predicateType == 'cosign.sigstore.dev/attestation/vuln/v1')].details.vulnerabilities[?(@.severity ==
      'Medium')] | length(@) }}
      - Низкие: {{ images[0].attestations[?(@.predicateType == 'cosign.sigstore.dev/attestation/vuln/v1')].details.vulnerabilities[?(@.severity == 'Low')] |
      length(@) }}
```

Сегодня рассмотрели

- что есть приложение
- почему devSECops это всё ещё сложно
- CR как интерфейс для пользователя
- как работает k8s-оператор
- что делать с приложением до скана
- что делать с отчётами после скана
- что делать с инструментами во время скана

Overall профит

⋯➧ **управление на уровне абстракций k8s**

- разгрузка очередей на сканирование
- выбор инструментов на основе характеристик артефакта
- сканирование на ресурсах разработчика
- унифицированный результат

Overall профит

- управление на уровне абстракций k8s

- **разгрузка очередей на сканирование**
- выбор инструментов на основе характеристик артефакта
- сканирование на ресурсах разработчика
- унифицированный результат

Overall профит

- управление на уровне абстракций k8s
- разгрузка очередей на сканирование

⇢ **выбор инструментов на основе характеристик артефакта**

- сканирование на ресурсах разработчика
- унифицированный результат

Overall профит

- управление на уровне абстракций k8s
- разгрузка очередей на сканирование
- выбор инструментов на основе характеристик артефакта

⋯➧ **сканирование на ресурсах разработчика**

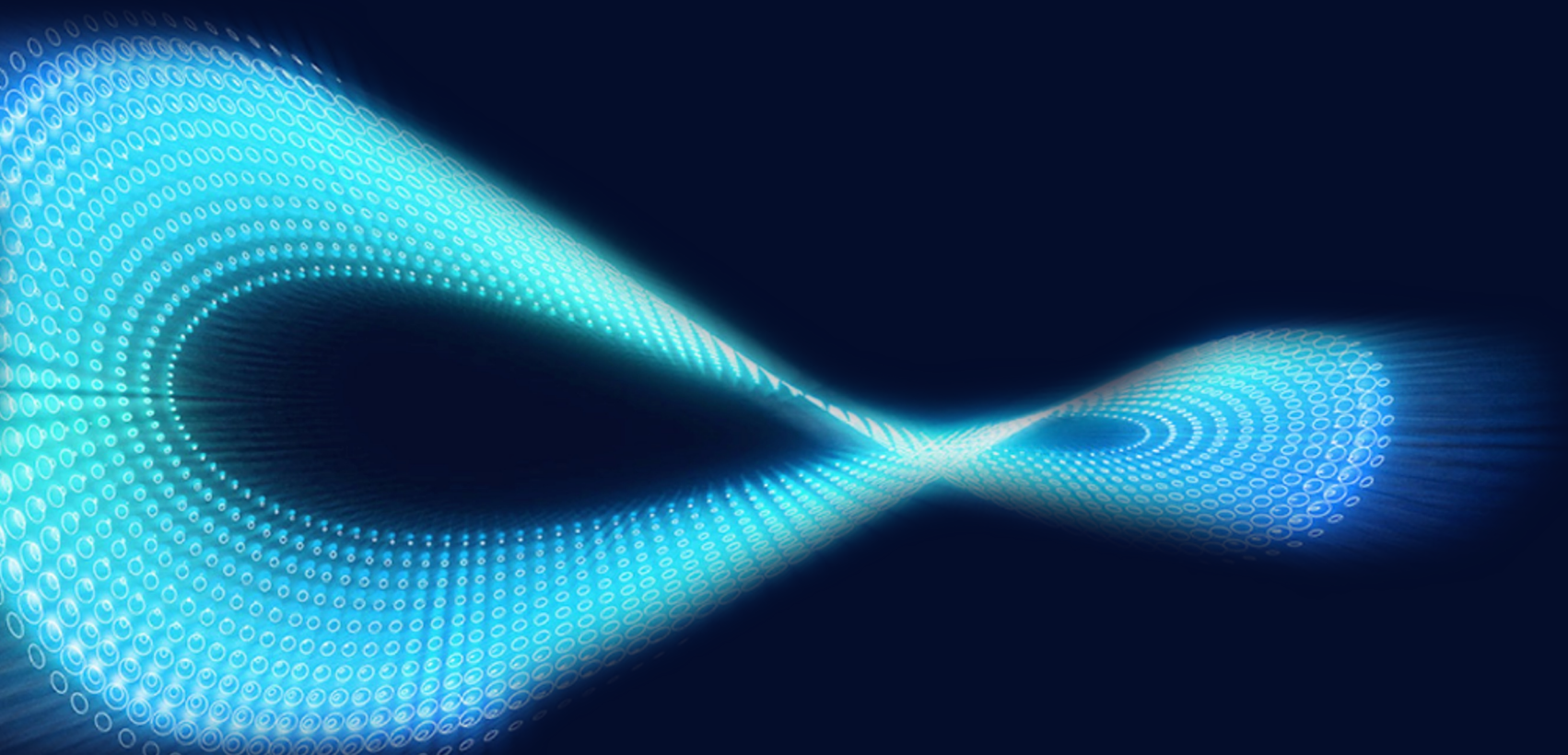
- унифицированный результат

Overall профит

- управление на уровне абстракций k8s
- разгрузка очередей на сканирование
- выбор инструментов на основе характеристик артефакта
- сканирование на ресурсах разработчика

⇢ **унифицированный результат**

Вопросы?



Алена Жилина

DevSecOps-инженер

Вячеслав Давыдов

DevSecOps-инженер