



KubeCon

THE LINUX FOUNDATION

**OPEN
SOURCE
SUMMIT**

China 2024



CloudNativeCon

 **AI_dev**
Open Source GenAI & ML Summit



KubeCon



CloudNativeCon



China 2024



经济增长安全 从构建到运行时的 SLSA 合规性实用指 南

2024年8月21日 安格罗 (*Enguerrand Allamel*) , *Ledger*

安格罗(*Enguerrand Allamel*)

- 学术经历：曾在清华大学学习一年
- 本人职位：Ledger 的高级云安全工程师
- 公司概述：Ledger 尖端于安全钱包硬件和尖端安全产品
- 当前关注：通过大力加强用户保护来增强供应链安全



LEDGER

1. 为什么供应链安全重要?
2. 什么是 SLSA (软件工件供应链级别) ?
3. 供应链安全防御的可能里程碑
4. 示例实现
 1. 在构建端
 2. 在途
5. 硬件安全模块 (HSM) 的深入探讨

供应链案例



KubeCon

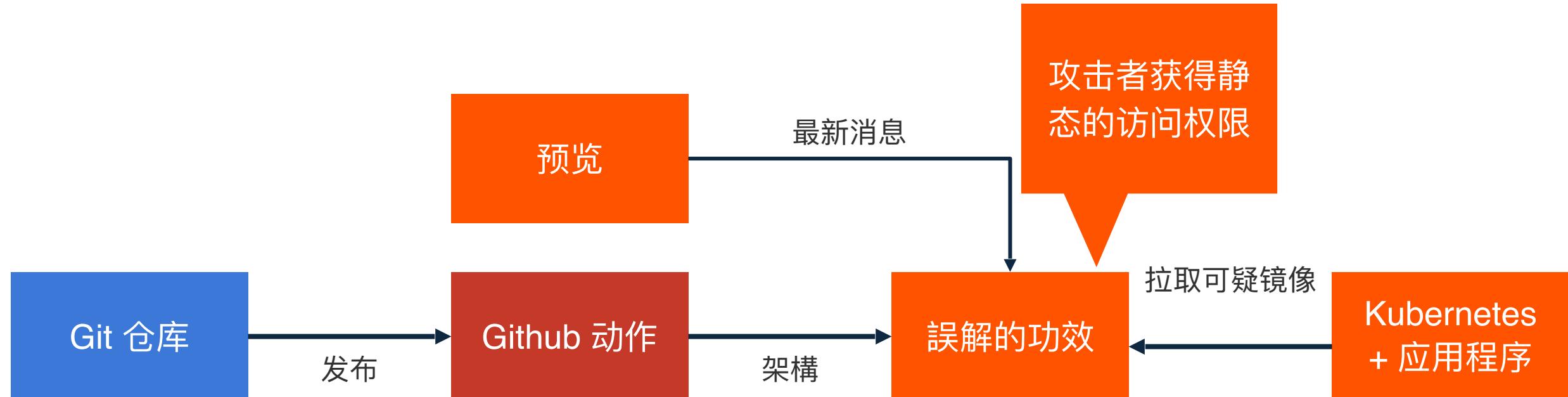


CloudNativeCon



THE LINUX FOUNDATION
OPEN SOURCE SUMMIT

China 2024



供应链安全为什么重要？



KubeCon



CloudNativeCon

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
China 2024

“高德纳预测到 2025 年，将有 45% 的组织经历过软件供应链攻击”*

攻击类型	案例研究
向源Git仓库提交未授权的更改	<p><u>SushiSwap</u>：具有存储库访问权限的承包商推送了恶意提交，将加密货币重定向到自身</p> <p><u>超过 300 万美元的用户资金受到影响</u></p>
破坏构建过程	<p><u>SolarWinds</u>：攻击者入侵了构建平台并安装了植入物，该植入物会在每次构建过程中注入恶意行为</p> <p><u>大规模数据泄露</u></p> <p><u>约有 18,000 个组织受到影响</u></p> <p><u>SolarWinds 股价下跌 40%</u></p>

什么是SLSA？



- **SLSA**: 软件工件的安全级别
- **支持**: 由 OpenSSF (开源安全基金会) 赞助, 与 Linux 基金会相关
- **协作框架**: 通过跨行业协作开发
- **目的**: 制定保护软件供应链的标准和指南
- **核心组件**:
 - SLSA 要求
 - SLSA 出处 (类似于证明)
- **受众**: 针对软件生产商、消费者和基础设施提供商

网站:

<https://slsa.dev/>

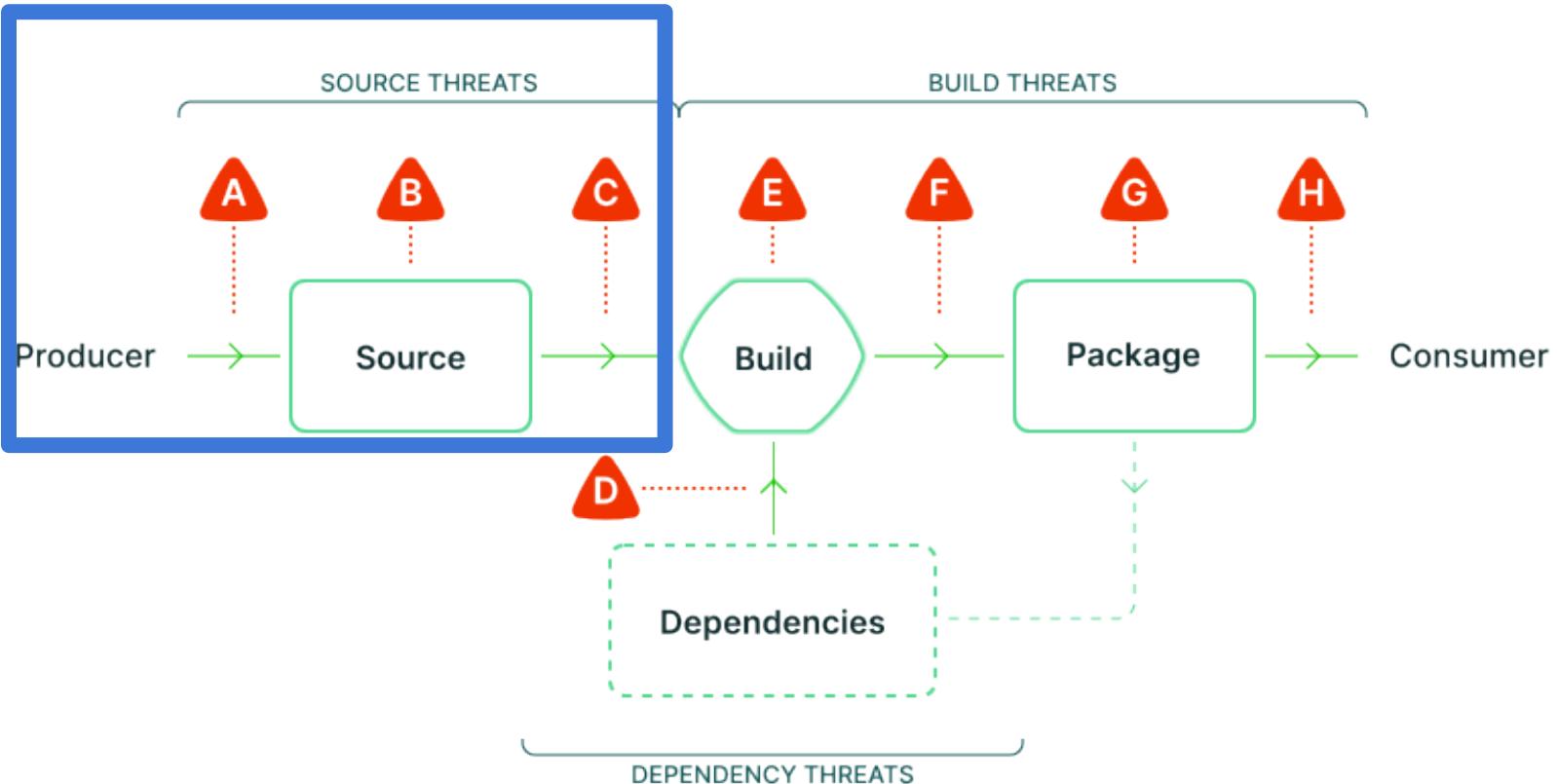
Github 存储库:

<https://github.com/slsa-framework/slsa>

SLSA 中的威胁和攻击范围：来源



China 2024



SOURCE THREATS

- A Submit unauthorized change
- B Compromise source repo
- C Build from modified source

DEPENDENCY THREATS

- D Use compromised dependency

BUILD THREATS

- E Compromise build process
- F Upload modified package
- G Compromise package registry
- H Use compromised package

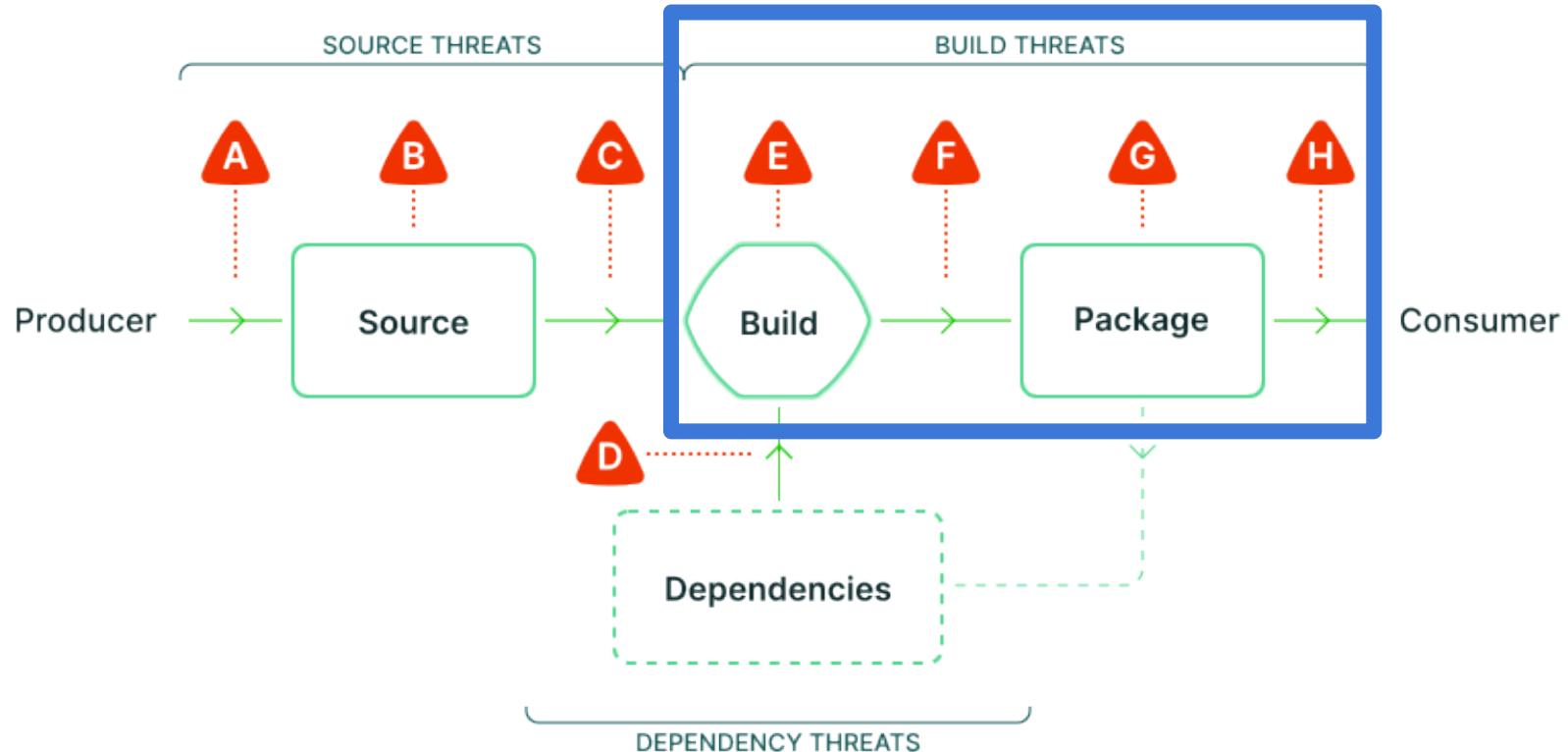
例子：

- Git 存储库内的代码修改
- Git 存储库托管平台（例如 GitLab、GitHub、Gitea）上的权限绕过

SLSA 中的威胁和攻击范围：构建



China 2024



例子：

- CI/CD 或构建平台受损
- 软件包注册表被盗

SOURCE THREATS

- A Submit unauthorized change
- B Compromise source repo
- C Build from modified source

DEPENDENCY THREATS

- D Use compromised dependency

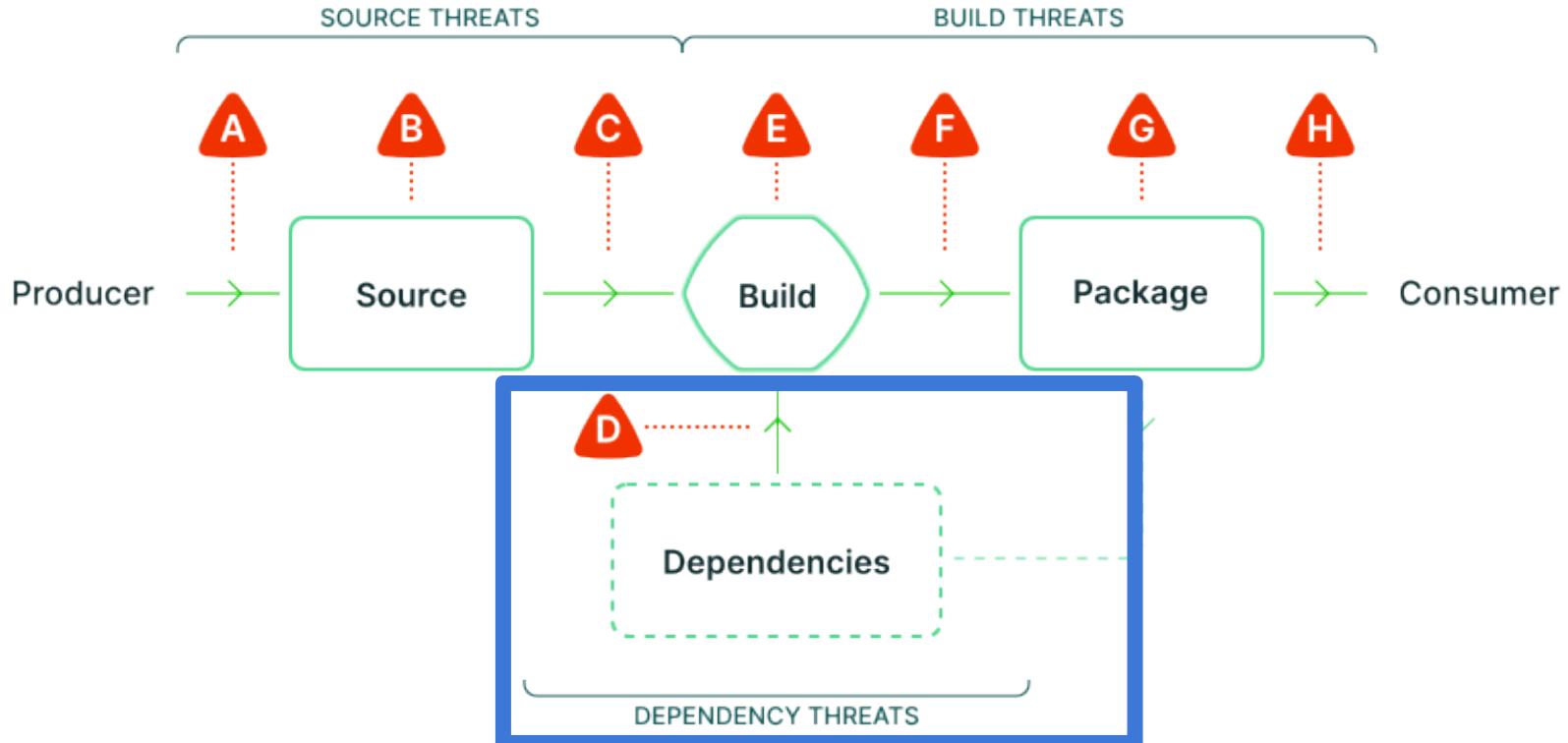
BUILD THREATS

- E Compromise build process
- F Upload modified package
- G Compromise package registry
- H Use compromised package

SLSA 中的威胁和攻击范围：依赖项



China 2024



SOURCE THREATS

- A Submit unauthorized change
- B Compromise source repo
- C Build from modified source

DEPENDENCY THREATS

- D Use compromised dependency

BUILD THREATS

- E Compromise build process
- F Upload modified package
- G Compromise package registry
- H Use compromised package

例子：

- 在 PyPI.org、npmjs.com 等平台上托管的依赖项中软件包的域名抢注。
- 依赖项中嵌入的恶意代码

供应链安全防御的可能里程碑： SLSA



KubeCon



CloudNativeCon

THE LINUX FOUNDATION
China 2024

安全级别定义链接至 SLSA 构建线程

目标复杂度	等级	要求	重点
默认情况下	<u>构建 L0</u>	(没有任何)	(不适用)
简单的	<u>构建 L1</u>	出处展示了软件包是如何构建的	错误、文档
简单至中等	<u>构建 L2</u>	由托管构建平台生成的签名出处	构建后篡改
难的	<u>构建 L3</u>	强化的构建平台	构建期间的篡改

1.0 版中的 SLSA 框架仅为构建威胁/轨迹定义级别。

表格基于 <https://slsa.dev/spec/v1.0/levels>

供应链安全防御的可能里程碑

命令	实践示例	重点
0: 默认	(没有任何)	(不适用)
1: 首次测试	本地第一个工件/证明签名，对 Kubernetes 集群上的镜像使用情况进行初步监控，审核当前 OSS 的使用/分发	测试防御机制和工具
2: 初始防御实施	在 CI/CD 管道内构建（非本地），在构建平台内签名工件/证明，在运行时构建 SBOM	建立供应链安全防御的基础层级
3: 高级防御	强化构建平台、OSS 注册代理、重建 OSS 工件、带有 CA 签名密钥的 HSM、专有无密钥工件签名等。	实施纵深防御并防范高级情况

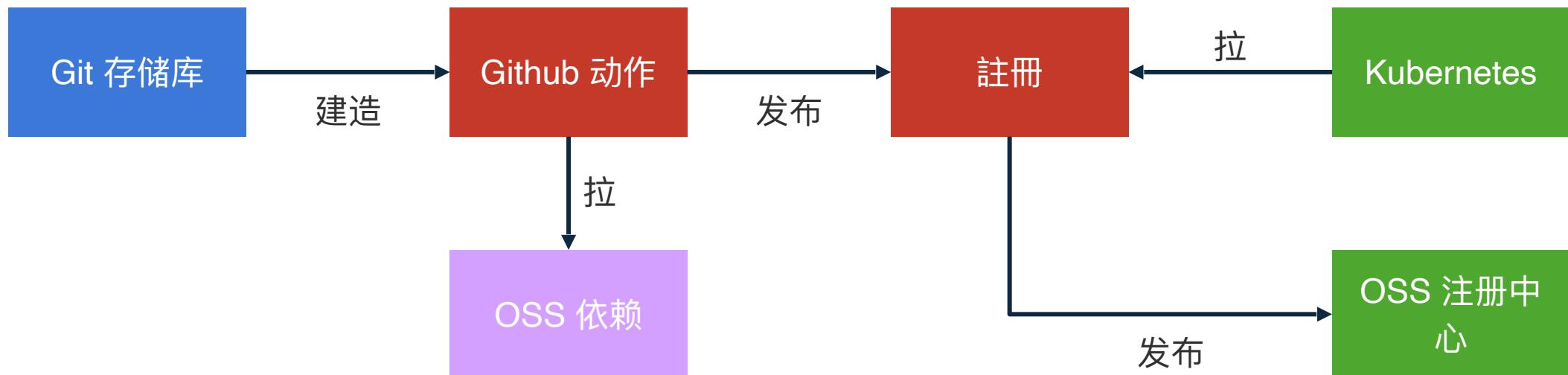
示例实现：上下文



China 2024

对于这种情况，我们假设以下设置：

- 应用程序在 Kubernetes 上运行
- 构建平台是 Github Action
- 使用开源 (OSS) 依赖项
- 开源 (OSS) 应用程序被构建并部署到公共注册中心



建设方面：可能的防御



KubeCon

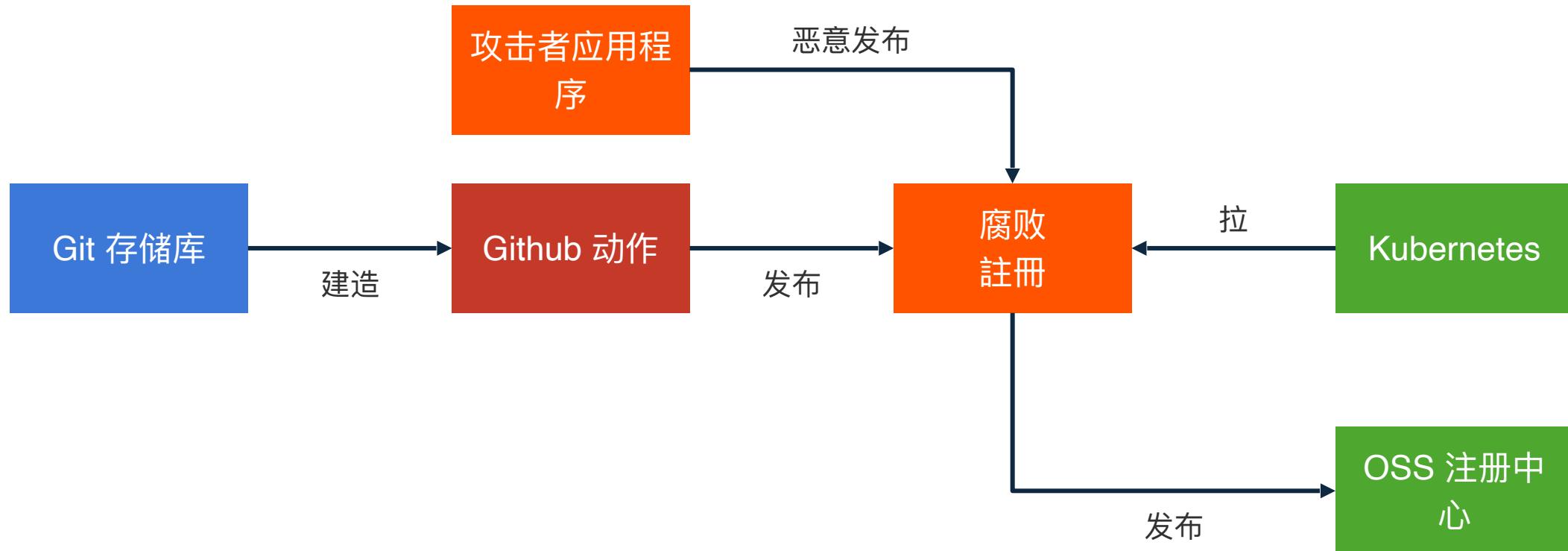


CloudNativeCon

THE LINUX FOUNDATION
OPEN SOURCE SUMMITAI_dev
Open Source Dev & ML Summit

China 2024

- 如何确保在 Kubernetes 上部署的软件是在 GitHub Actions 内构建的?
 - 签名：使用 Cosign、Notary 等工具。
 - **Provenance**：实现 SLSA Provenance、In-Toto 证明等。



构建方： Sigstore

- **Sigstore**：软件供应链安全的开源项目
- 支持：由 OpenSSF（开源安全基金会）赞助
- 目的：提供一种简单、安全的软件签名方法
- 座右铭：“签名、验证、保护”
- 核心功能：工件签名、签名验证和监控
- 支持的格式：适用于 blob、容器图像等。
- 提供的工具：**Cosign**：用于签名的命令行界面 (CLI)、**Fulcio**：无密钥签名机构、**Rekor**：透明元数据日志记录等



文档：

<https://docs.sigstore.dev/>

Github 组织：

<https://github.com/sigstore>

构建方面：无密钥签名与静态签名



KubeCon



CloudNativeCon

THE LINUX FOUNDATION
OPEN SOURCE SUMMITAI_dev
Open Source Dev & ML Summit

China 2024

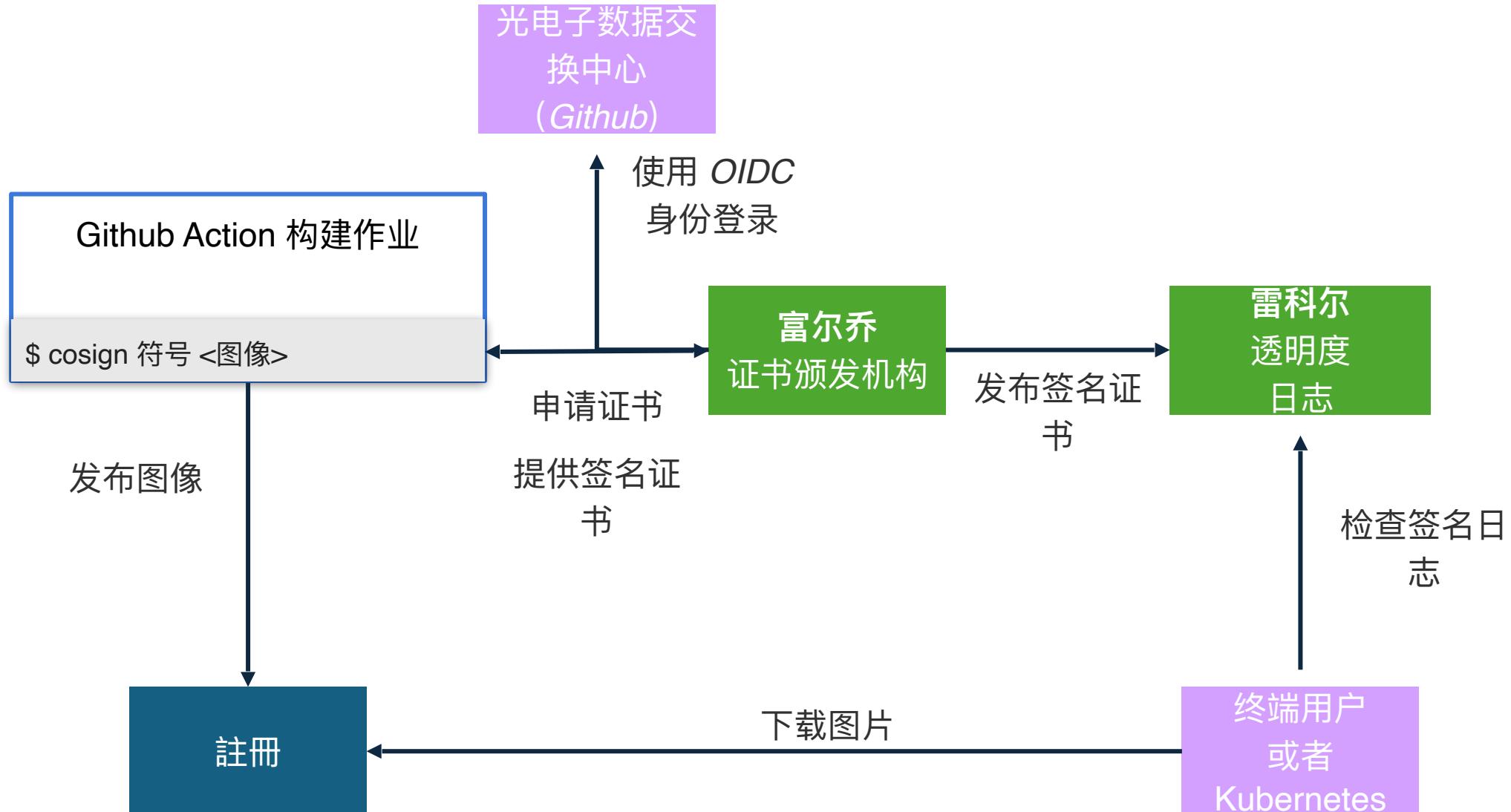
类型	描述	Cosign CLI 命令
静止的	生成的私钥/公钥是使用、自我管理的，不基于 OIDC	\$ cosign sign --key cosign.key myimage:v1
无钥匙	根据 OIDC 身份（例如 GitHub、Google、Microsoft）使用临时密钥	\$ cosign 签署 myimage:v1

默认建议：默认使用无密钥签名，建议使用以增强安全性和透明度

构建方面：签名无钥匙



China 2024



构建端：Github内部签名操作

- **CI/CD 集成：**在 CI/CD 管道中，特别是在 GitHub Actions 中的构建作业期间，容器镜像会被签名
- **超越签名：**单靠签名是不够的，证明可以提供额外的信息来增强安全性

...
工作：
构建并推送：
步骤：
-名称：安装 Cosign
 用途：sigstore/cosign-installer@v3
...
-名称：加载Docker元数据
 用途：docker/metadata-action@v5
...
-名称：构建并推送容器镜像
 用途：docker/build-push-action@v6
 id : 构建并推送
...
- name : 使用 GitHub OIDC Token 对图像进行签名
环境：
 摘要：\${{ steps.build-and-push.outputs.digest }}
 标签：\${{ steps.docker_metadate.outputs.tags }}
运行：
图片=""
对于 \${TAGS} 中的标签；执行
图片+="\${tag}@\${DIGEST}"
...
...
...

构建方： In-Toto 证明

- **In-Toto**: 保护供应链完整性的开源框架
- 支持: 由 CNCF 赞助
- 目的: 提高软件供应链的透明度和安全性
- 全球范围: 专注于供应链安全, 集成多种语言, 主要是 Python
- **SLSA 集成**: 可以纳入 SLSA Provenance 规范
- **详细证明**: 提供关键的供应链信息, 例如代码测试结果或代码审查证明



网站:

<https://in-toto.io/>

演示 (全局项目) :

[https://github.com/in-toto/
demo](https://github.com/in-toto/demo)

认证规范:

[https://github.com/in-toto/
attestation/tree/v1.0/](https://github.com/in-toto/attestation/tree/v1.0/)

构建方： In-Toto 证明：示例



KubeCon



CloudNativeCon

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT

China 2024

- 谓词文件：嵌入在证明中的元数据或信息
 - 示例：测试结果、运行器详细信息、构建环境等。
- Cosign 集成：Cosign 可以创建和签署谓词文件，类似于处理容器或 blob 的方式
- 增强的安全性：向软件消费者提供可信的信息
 - 示例：证明测试已通过或代码已审查

```
{  
  "_type": "https://in-toto.io/Statement/v0.1",  
  "predicateType": "https://cosign.sigstore.dev/attestation/v1",  
  "主题": [  
    {  
      "名称": "ghcr.io/ledgerhq/signed-image",  
      "摘要": {  
        "sha256": "<图像-sha256>"  
      }  
    }  
  ],  
  "谓词": {  
    "<我的数据>": "<我的值>",  
    "时间戳": "2021-08-11T14: 51: 09Z"  
  }  
}
```

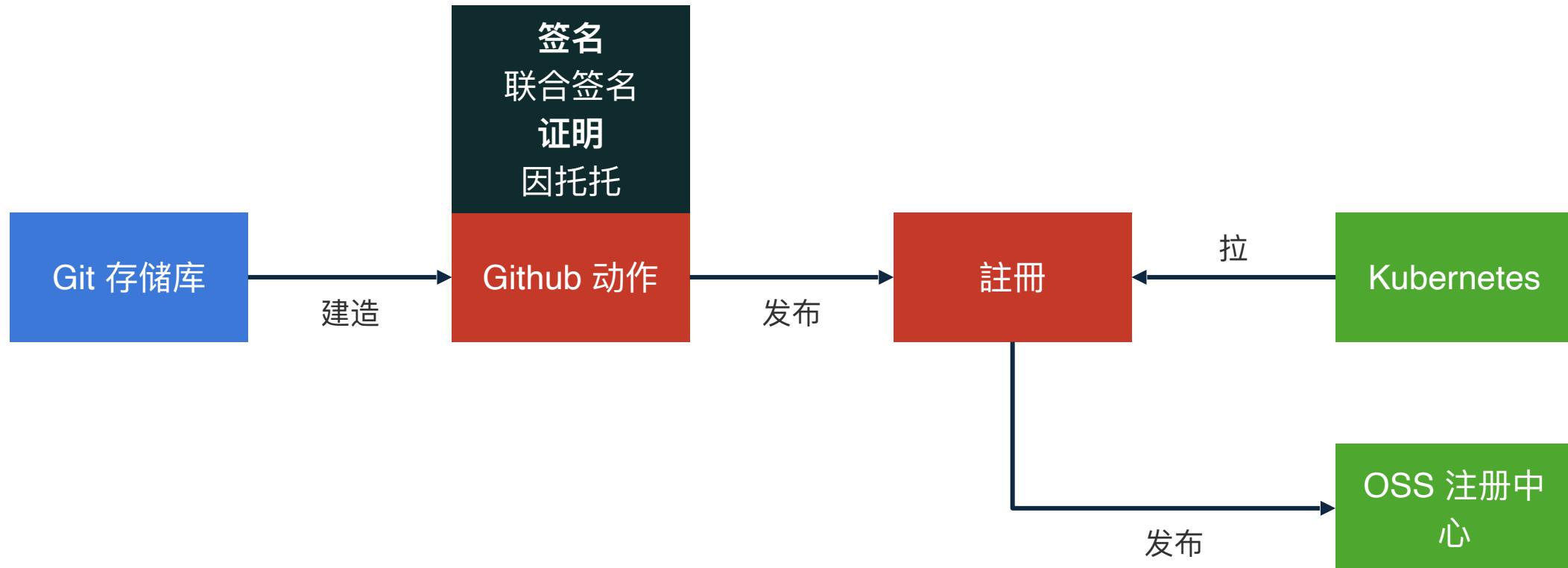
```
$ cosign attest--predicate <文件> <图像>  
$ cosign verify-attestation <图像>
```

构建方面：概述



China 2024

- 签名：使用 Cosign 在构建运行器中执行
- 证明：在构建运行器中使用 Cosign 与 In-Toto 结合执行
- 可信信息：通过提供可验证的详细信息确保构建和工件的完整性

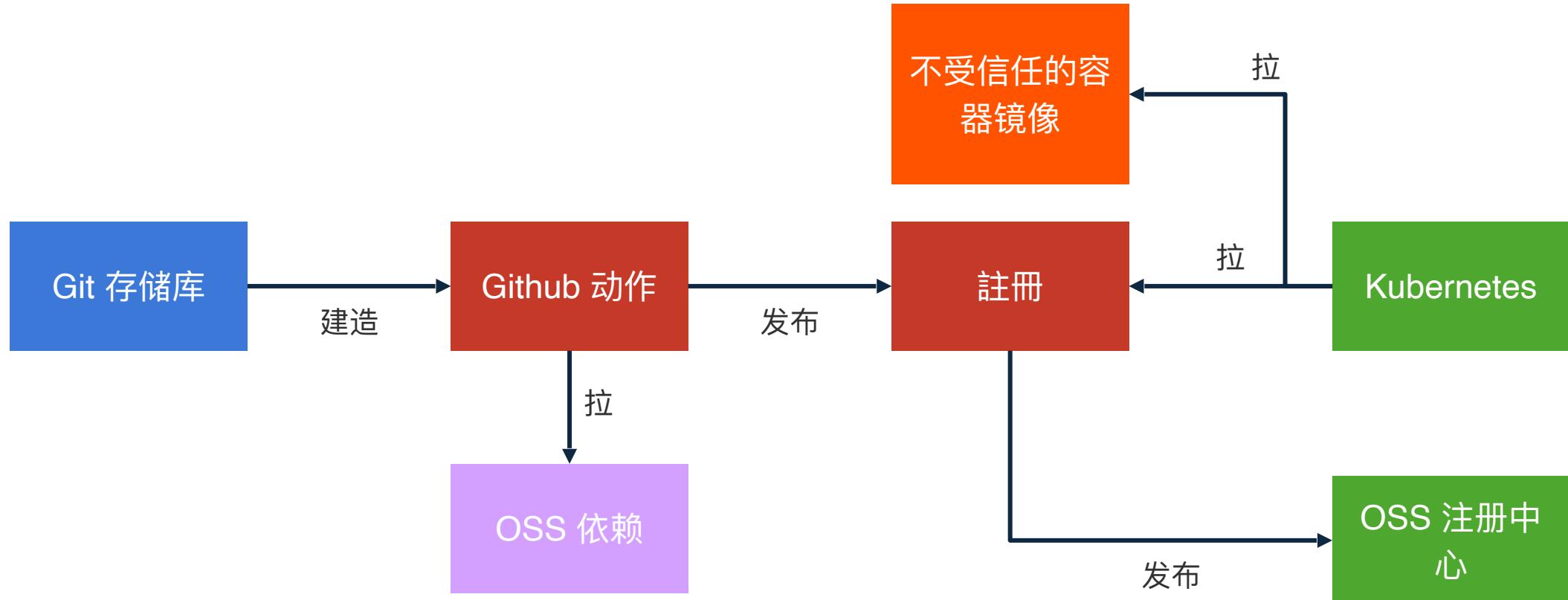


运行时方面：可能的防御



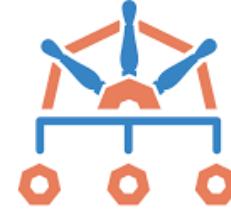
China 2024

- 如何验证生产中运行的应用程序来源?
 - **Kubernetes** 准入控制器: Cosign 策略控制器、Kyverno 等。
 - 审计与检测: Kubescape等



运行时端： Kyverno

- **Kyverno** : Kubernetes 的开源策略引擎
- 支持: 由 CNCF 赞助
- 目的: 在 Kubernetes 中执行安全性、合规性和运营政策
- 功能: 验证和清理 Kubernetes 资源、审计和报告策略等
- 策略格式: 策略以 Kubernetes 资源的形式编写
- 供应链安全执行: 确保仅部署经过正确签名和认证的图像



Kyverno

网站:

<https://kyverno.io/>

Github 组织:

<https://github.com/kyverno>

示例政策链接至供应链安全:

[https://kyverno.io/
policies/?policytypes=Software
SupplyChain%2520Security](https://kyverno.io/policies/?policytypes=SoftwareSupplyChain%2520Security)

运行时端：验证



KubeCon



CloudNativeCon



THE LINUX FOUNDATION
OPEN SOURCE SUMMIT



China 2024

- 签名验证：检查容器镜像签名的策略
- 证明验证：验证证明内容的政策
- 示例：确保所有与正则表达式匹配的图像都使用正确的密钥进行签名

apiVersion: kyverno.io/v1

种类: ClusterPolicy

元数据:

名称: 验证图像

规格:

validationFailureAction : 强制

规则:

-名称: 验证图像

匹配:

任何:

-资源:

种类:

- 吊舱

验证图像:

- 图片参考:

- “ghcr.io/ledgerhq/signed-*”

证明人:

- 参赛作品:

无钥匙:

主题: “https://<url-to-the-workflow> @ <refs>”

发行人: “https://token.actions.githubusercontent.com”

运行时端：Kubescape

- **Kubescape**：开源 Kubernetes 安全平台
- 支持：由 CNCF（云原生计算基金会）赞助并与 Linux 基金会挂钩
- 全球范围：专注于增强 Kubernetes、CI/CD 管道和源代码的安全性
- 功能：包括 Kubernetes 扫描器、CI/CD 集成等



Kubescape

网站：

<https://kubescape.io/>

Github 组织：

<https://github.com/kubescape>

运行时方面：分析你的Kubernetes集群



KubeCon



CloudNativeCon



THE LINUX FOUNDATION
OPEN SOURCE SUMMIT



AI_dev
Open Source DevOps & ML Summit

China 2024

- 扫描执行：根据预定义的控件运行扫描
- 模式：扫描可以一次性执行，也可以连续执行
- 注册表用法：
 - 确定受信任映像注册表的使用情况
 - 检测不安全的图像注册表的使用情况（良好的第一步）
- 图像签名验证：
 - 检查图像签名是否存在（良好的第一步）
 - 验证图像签名的真实性

附加控件：

文档中提供了更多控件：
<https://hub.armosec.io/docs/controls>

运行时方面：分析你的Kubernetes集群



China 2024

控制： C-0237：检查签名是否存在 (<https://hub.armosec.io/docs/c-0237>)

```
$ kubescape 扫描控制“C-0237” -v
```

```
...  
#####
```

```
Api版本: apps/v1
```

```
类型: 部署
```

```
名称: my-hello-ledger
```

```
命名空间: default
```

```
控制: 1 (失败: 1, 所需操作: 0)
```

严重程度	控制名称	文档	辅助补救
------	------	----	------

高	检查签名是否存在	https://hub.armosec.io/docs/c-0237	spec.template.spec.containers[0].image
---	----------	---	--

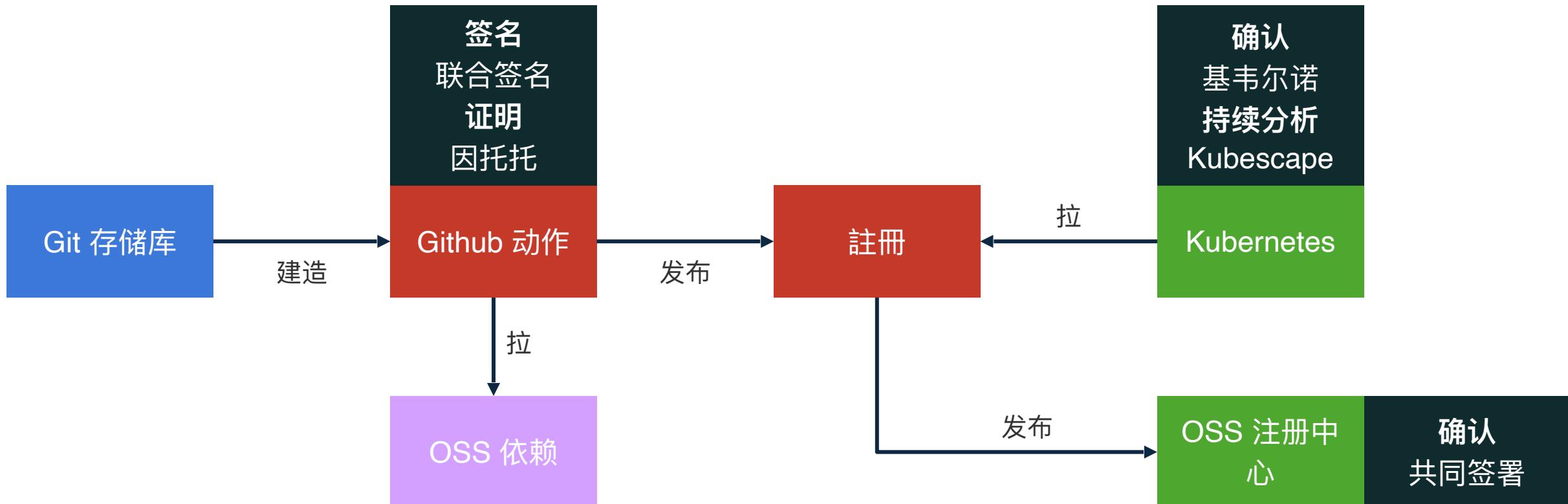
严重性	控制名称	失败的资源	所有资源	合规性分数
-----	------	-------	------	-------

本实施概述



China 2024

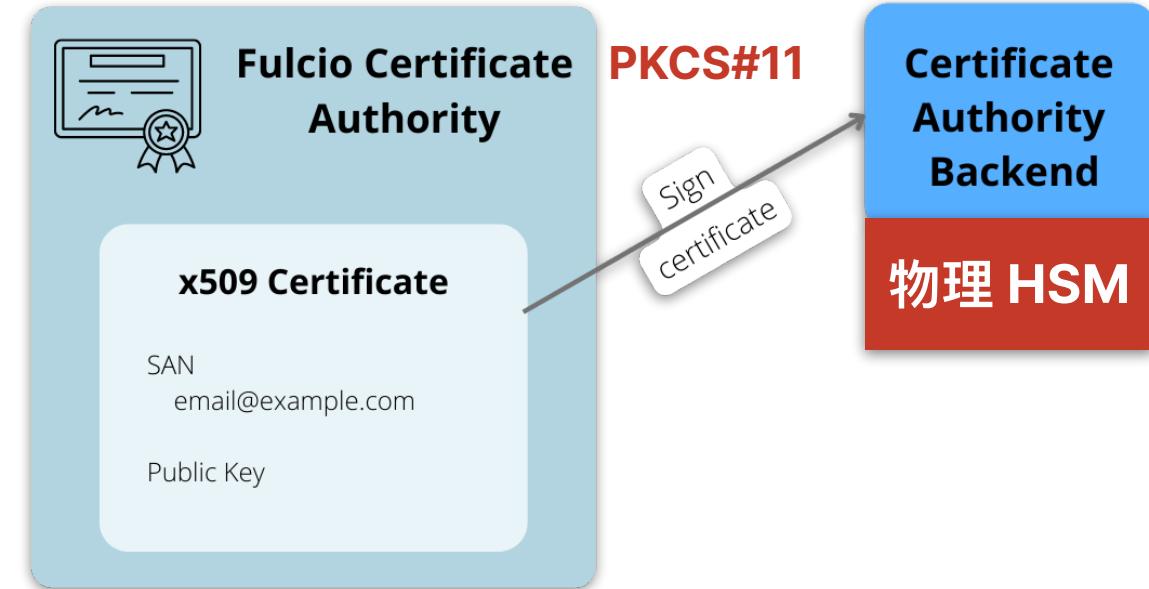
- 签名：使用 Cosign 在构建运行器中执行
- 证明：在构建运行器中使用 Cosign 与 In-Toto 结合执行
- 验证：在 Kubernetes 集群内进行，以验证容器镜像的完整性，或者在本地使用 Cosign



进一步了解HSM（硬件安全模块）



- **HSM（硬件安全模块）**：专为加密操作而设计的物理设备
- **私有证书保护**：HSM 提供高级别的物理安全性来保护私有证书
- **证书颁发机构 (CA)**：持有根证书，通过 Fulcio 使用
- **Fulcio**：充当构建系统和证书颁发机构之间的纽带
- **Sigstore Stack**：包括 Fulcio 和 Rekor 在内的完整堆栈可以托管在 Kubernetes 上，以形成一个独立的解决方案
- **隐私注意事项**：签署私人物品时，使用公共 Fulcio 和 Rekor 服务可能会泄露有关您的签名的信息



问题?



KubeCon



CloudNativeCon



China 2024



- 您有任何问题或意见吗?
- 附加资源:
 - CNCF 标签安全白皮书: [https://
project.linuxfoundation.org/hubfs/
CNCF_SSCP_v1.pdf](https://project.linuxfoundation.org/hubfs/CNCF_SSCP_v1.pdf)