

Driving Sustainability with Platform Engineering



Lesley Cordero, @clesleycode
The New York Times, Staff Engineer

Hi! I'm **Lesley Cordero**.

Fan of boba, dogs, and sustainable organizations.

My experience leading Platform Engineering teams has me thinking about **how developer platforms can drive organizational sustainability**.



Driving Sustainability with Platform Engineering

The continuous practice of operating in a way that enables short-term growth opportunities while enabling long-term success.

Driving Sustainability with Platform Engineering

The continuous practice of operating in a way that enables short-term growth opportunities while enabling long-term success.

Driving Sustainability with Platform Engineering

The continuous practice of operating in a way that enables short-term growth opportunities while enabling long-term success.

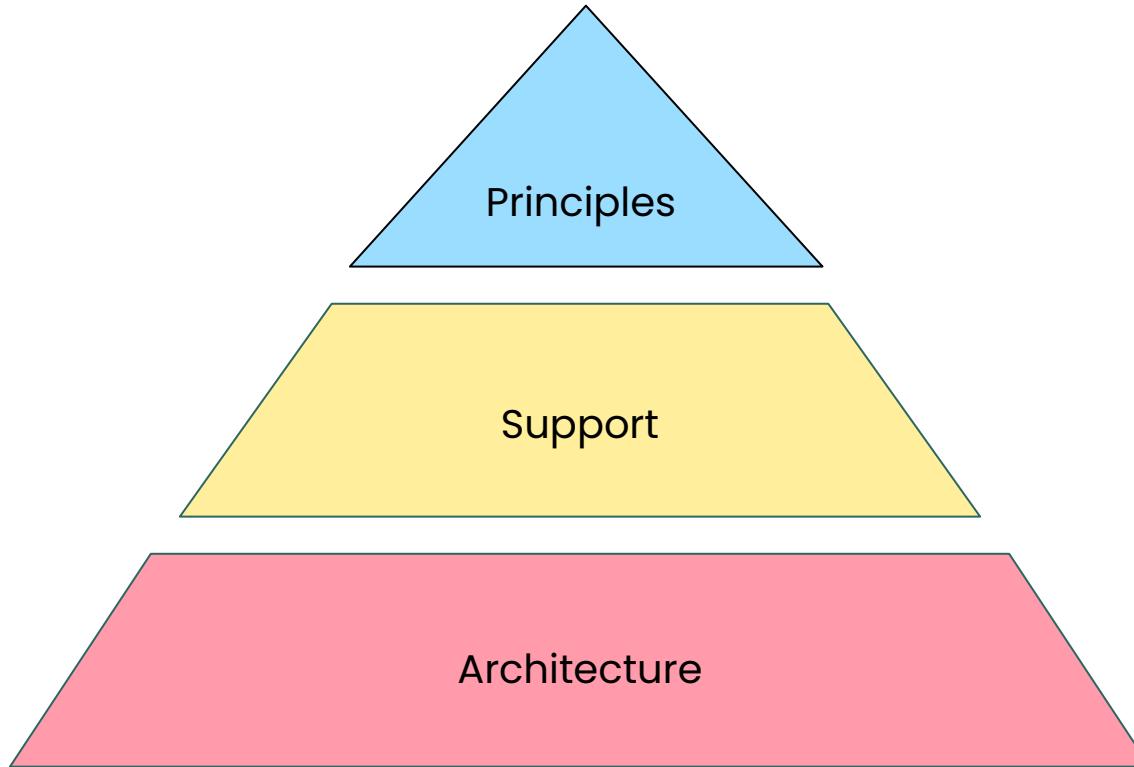
Driving Sustainability with Platform Engineering

The continuous practice of operating in a way that enables short-term growth opportunities while enabling long-term success.

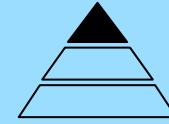
Defining Platform Engineering

Platform Engineering drives sustainability by practicing **sociotechnical principles** that provide a **strong support system** for application developers using our **standardized shared platform architecture**.

Defining a “Platform”



Platform Principles



Principles
Support
Architecture

Defining our platform principles

1. **Culture**
2. **Automation**
3. **Lean**
4. **Measurement**
5. **Sharing**



DevOps



Platform Engineering

Defining our platform principles

1. **Culture**
2. **Automation**
3. **Lean**
4. **Measurement**
5. **Sharing**



DevOps



Platform Engineering

Defining our platform principles

- 1. Culture**
2. Automation
3. Measurement



DevOps

DevOps drives a culture of continuous improvement and reduces knowledge silos by intentionally sharing knowledge & feedback.



Platform Engineering

Platform Engineering drives a culture of continuous improvement through a community based approach to building software & knowledge sharing.

Defining our platform principles

- 1. Culture**
2. Automation
3. Measurement



DevOps

DevOps drives a culture of continuous improvement and reduces knowledge silos by intentionally sharing knowledge & feedback.



Platform Engineering

Platform Engineering drives a culture of continuous improvement through a **community based approach** to building software & knowledge sharing.

Defining our platform principles

1. Culture
2. Automation
3. Measurement



DevOps

DevOps drives a culture of continuous improvement and reduces knowledge silos by intentionally sharing knowledge & feedback.



Platform Engineering

Platform Engineering drives a culture of continuous improvement through a **community based approach** to building software & knowledge sharing.

Defining our platform principles

1. Culture
2. **Automation**
3. Measurement



DevOps

Improves efficiency, reduces human error, and enables faster delivery by automating manual tasks.



Platform Engineering

Improves efficiency and enables faster software delivery by automation and consolidating development efforts.

Defining our platform principles

1. Culture
2. **Automation**
3. Measurement



DevOps

Improves efficiency, reduces human error, and enables faster delivery by automating manual tasks.



Platform Engineering

Improves efficiency and enables faster software delivery by automation and consolidating development efforts.

Defining our platform principles

1. Culture
2. **Automation**
3. Measurement



DevOps

Improves efficiency, reduces human error, and enables faster delivery by automating manual tasks.



Platform Engineering

Improves efficiency and enables faster software delivery by automation and consolidating development efforts.

Defining our platform principles

1. Culture
2. Automation
3. **Measurement**



DevOps

Identify areas of improvement by regularly evaluating the software delivery process through qualitative and quantitative measures.



Platform Engineering

Feedback-driven approach to identifying areas of improvement through the software engineering process.

Defining our platform principles

1. Culture
2. Automation
3. **Measurement**



DevOps

Identify areas of improvement by regularly evaluating the software delivery process through qualitative and quantitative measures.



Platform Engineering

Feedback-driven approach to identifying areas of improvement through the software engineering process.

Platform Support



Principles
Support
Architecture

Platform Support Strategies



1. Make building sustainable software easy

Reduce the work needed to build software with a minimum bar of quality, capable of continuous improvement.

Platform Support Strategies



1. Make building sustainable software easy

Reduce the work needed to build software with a minimum bar of quality, capable of continuous improvement.

2. Scalable, flexible adoption techniques

Support engineers through scalable techniques, while providing avenues for high-touch support.

Platform Support Strategies



1. Make building sustainable software easy

Reduce the work needed to build software with a minimum bar of quality, capable of continuous improvement.

2. Scalable, flexible adoption techniques

Support engineers through scalable techniques, while providing avenues for high-touch support.

3. Support engineers in building their skills

By removing cognitive load, product engineers can reallocate saved time to building skills that can't be automated.

Platform Support Strategies



1. Make building sustainable software easy

Reduce the work needed to build software with a minimum bar of quality, capable of continuous improvement.

2. Scalable, flexible adoption techniques

Support engineers through scalable techniques, while providing avenues for high-touch support.

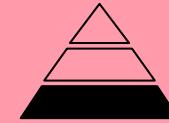
3. Support engineers in building their skills

By removing cognitive load, product engineers can reallocate saved time to building skills that can't be automated.

4. Drive a sustainable culture

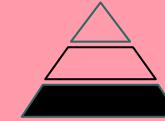
Use position as platform engineers to drive a culture that reduces burnout of all kinds.

Platform Architecture



Principles
Support
Architecture

Architecture Principles



Principles
Support
Architecture

1. Design driven architecture

Use architecture design best practices to build our developer platform.

Architecture Principles



1. Design driven architecture

Use architecture design best practices to build our developer platform.

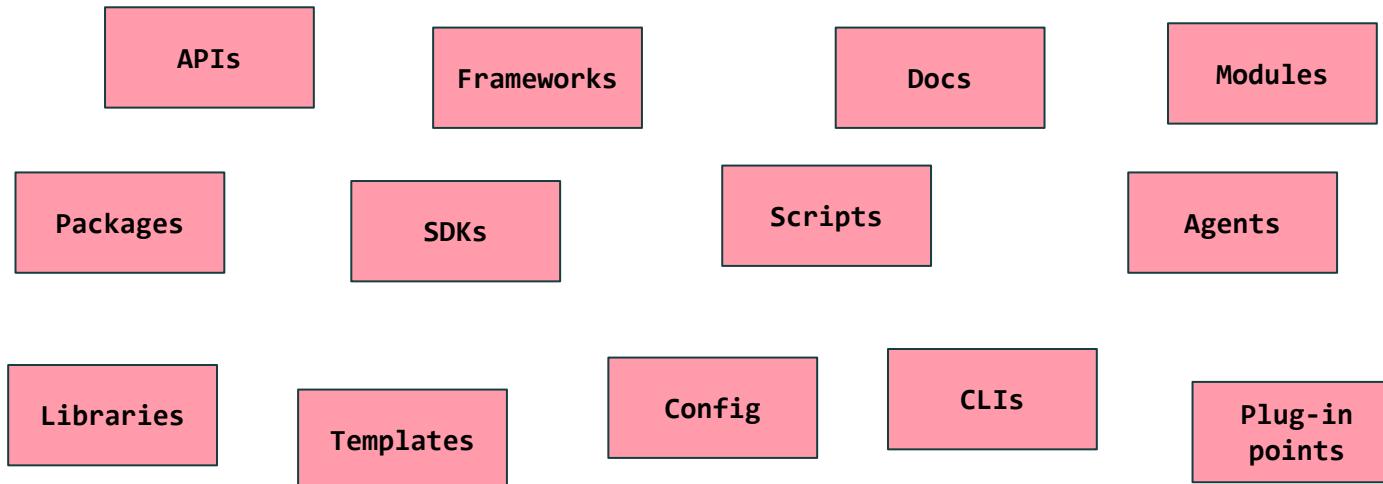
2. Responsive to evolving architecture needs

Build a platform that complements product architecture needs, avoiding platforms that are ultimately not useful.

Chaos



Principles
Support
Architecture



Design Tensions



Principles
Support
Architecture

1. Standardization & Flexibility

Since our platform is shared, standardization is crucial to reducing drift, but has the potential to be too opinionated.

Design Tensions



Principles
Support
Architecture

1. Standardization & Flexibility

Since our platform is shared, standardization is crucial to reducing drift, but has the potential to be too opinionated.

2. Simplicity & Complexity

As you respond to the evolving needs of your users, complexity becomes harder to manage.

Design Tensions



Principles
Support
Architecture

1. Standardization & Flexibility

Since our platform is shared, standardization is crucial to reducing drift, but has the potential to be too opinionated.

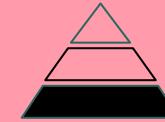
2. Simplicity & Complexity

As you respond to the evolving needs of your users, complexity becomes harder to manage.

3. Integrations without coupling

Integration points are powerful productivity techniques, but inject the risk of complexity.

Design Tensions



Principles
Support
Architecture

1. Standardization & Flexibility

Since our platform is shared, standardization is crucial to reducing drift, but has the potential to be too opinionated.

2. Simplicity & Complexity

As you respond to the evolving needs of your users, complexity becomes harder to manage.

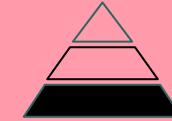
3. Integrations without coupling

Integration points are powerful productivity techniques, but inject the risk of complexity.

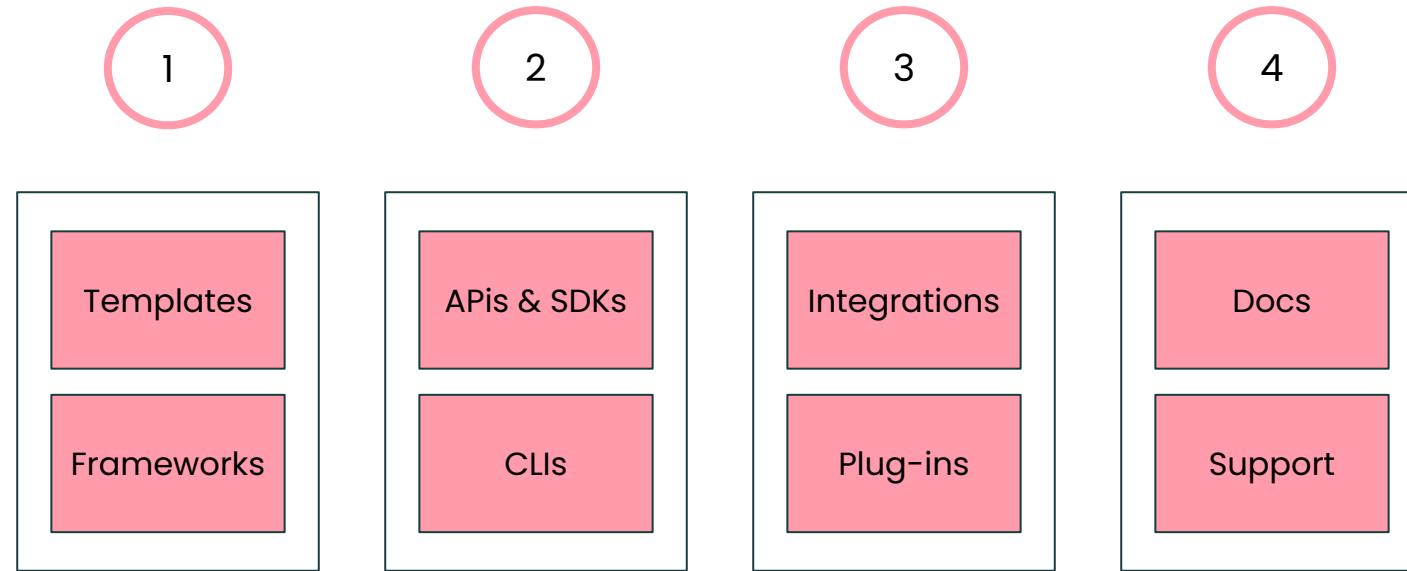
4. Build, Buy, or Reuse

Deciding when to take on the risk of maintaining your own tools, vendor lock-in, and relying on open source tooling.

Platform “Recipe”



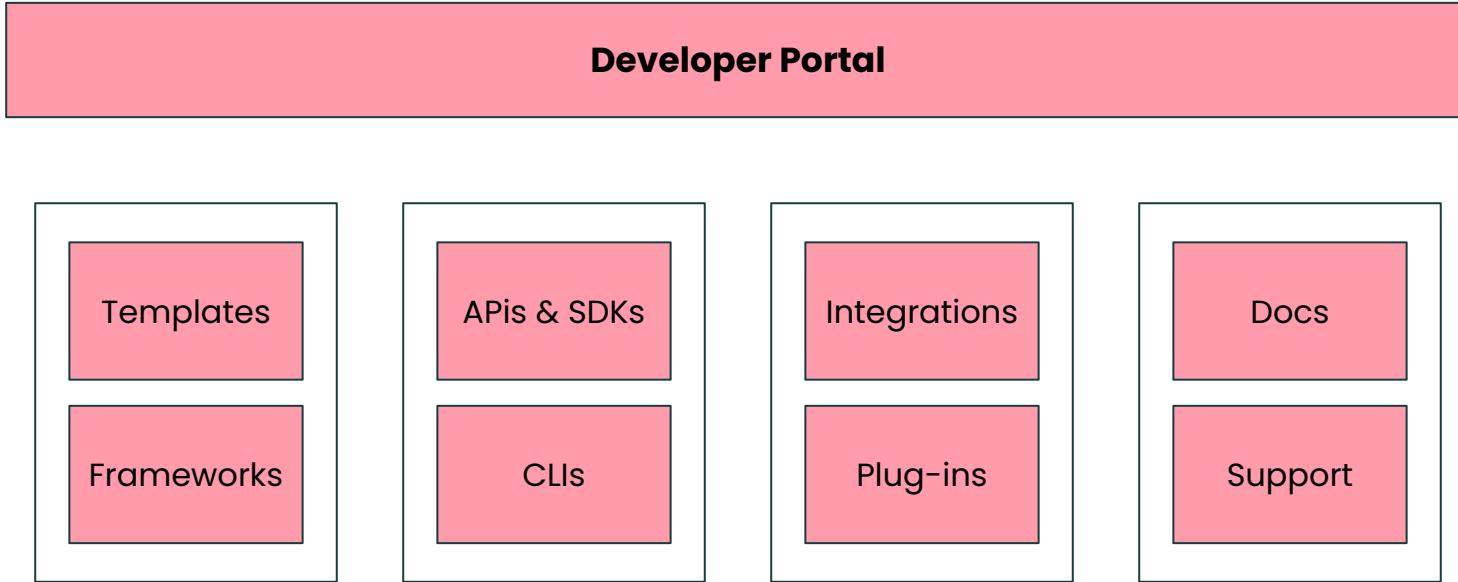
Practices
Support
Architecture



Platform “Recipe”



Practices
Support
Architecture



Feature Development Cycle



Requirements	Engineering Design	Development	Prod Readiness	Release	Maintenance
Partnering with product management to define feature requirements.	Designing the software that will enable your release.	The actual implementation of your feature.	Preparing for the challenges that might come from production.	Finally, release! 🎉	The ongoing cost of maintaining the feature in production.



Development

Supports the end-to-end development cycle, particularly runtime and architecture needs.



Delivery

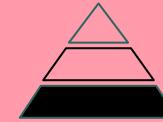
Supports the delivery or deployment of software by focusing on infrastructure management.



Reliability

Supports the post-delivery needs of software through production readiness techniques.

Development



Principles
Support
Architecture

Runtime Templates

Runtime templates minimize the amount of boilerplate code engineers write.

Development



Principles
Support
Architecture

Runtime Templates

Runtime templates minimize the amount of boilerplate code engineers write.

```
import express, { Request, Response } from
'express';

const app = express();
const port = 3000;

app.use(express.json());

app.get('/', (req: Request, res: Response) =>
{
    res.send('Hello, World!');
});

app.listen(port, () => {
    console.log(`Server is running on port
${port}`);
});
```

Development



Principles
Support
Architecture

Runtime Templates

Runtime templates minimize the amount of boilerplate code engineers write.

Runtime Packages

Complementary to templates, these packages provide common libraries to enable flexibility for applications.

Development



Principles
Support
Architecture

Runtime Templates

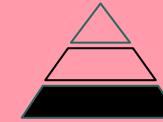
Runtime templates minimize the amount of boilerplate code engineers write.

Runtime Packages

Complementary to templates, these packages provide common libraries to enable flexibility for applications.

```
function standardHeader(req: Request, res: Response, next: NextFunction) {  
  res.setHeader('X-Standard-Header', 'DefaultValue');  
  next();  
}
```

Development



Principles
Support
Architecture

Runtime Templates

Runtime templates minimize the amount of boilerplate code engineers write.

Runtime Packages

Complementary to templates, these packages provide common libraries to enable flexibility for applications.

Package Manager

Provides access & visibility into your ecosystem of tools.

Development



Runtime Templates

Runtime templates minimize the amount of boilerplate code engineers write.

Runtime Packages

Complementary to templates, these packages provide common libraries to enable flexibility for applications.

Package Manager

Provides access & visibility into your ecosystem of tools.

Interface Documentation

Available interfaces should be well documented for ease of integration.

Delivery



Principles
Support
Architecture

Config Templates

Config templates minimize the amount of configuration engineers need to manage.

Delivery



Principles
Support
Architecture

Config Templates

Config templates minimize the amount of configuration engineers need to manage.

```
deploymentName: my-app-deployment
appName: my-app
containerName: my-app-container
imageName: my-app-image:1.0.0
containerPort: 8080
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Values.deploymentName }}
spec:
  selector:
    matchLabels:
      app: {{ .Values.appName }}
  template:
    metadata:
      labels:
        app: {{ .Values.appName }}
    spec:
      containers:
        - name: {{ .Values.containerName }}
          image: {{ .Values.imageName }}
          ports:
            - containerPort: {{ .Values.containerPort }}
```

Delivery



Principles
Support
Architecture

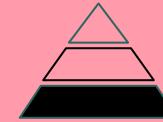
Config Templates

Config templates minimize the amount of configuration engineers need to manage.

Modularized Templates

Organize templates into reusable modules, using modularization techniques.

Delivery



Principles
Support
Architecture

Config Templates

Config templates minimize the amount of configuration engineers need to manage.

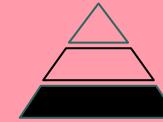
Modularized Templates

Organize templates into reusable modules, using modularization techniques.

CI/CD Integrations

Use automated pipelines that minimize exposure to underlying infrastructure templates.

Delivery



Principles
Support
Architecture

Config Templates

Config templates minimize the amount of configuration engineers need to manage.

Modularized Templates

Organize templates into reusable modules, using modularization techniques.

CI/CD Integrations

Use automated pipelines that minimize exposure to underlying infrastructure templates.

Guidance & Best Practice Documentation

Enable engineers to apply their domain expertise to operations work that requires business context.

Reliability



Principles
Support
Architecture

Config Templates

Extend config templates to include features that enable application reliability.

Reliability



Principles
Support
Architecture

Config Templates

Extend config templates to include features that enable application reliability.

```
opentelemetry:  
  tracing:  
    enabled: true  
    exporter:  
      jaeger:  
        endpoint: http://jaeger-collector:14250  
    propagation:  
      b3:  
        enabled: true
```

Reliability



Principles
Support
Architecture

Config Templates

Extend config templates to include features that enable application reliability.

Runtime Packages

Extend package ecosystem with modules that bake reliability into the way they were built.

Reliability



Principles
Support
Architecture

Config Templates

Extend config templates to include features that enable application reliability.

Runtime Packages

Extend package ecosystem with modules that bake reliability into the way they were built.

```
export const ollyMiddleware = (_req: Request, _res: Response, next: NextFunction): void => {
  const currentSpan = trace.getSpan(context.active());

  if (currentSpan) {
    currentSpan.setAttribute('example', 'example1');
  }
  next();
};
```

Reliability



Principles
Support
Architecture

Config Templates

Extend config templates to include features that enable application reliability.

Runtime Packages

Extend package ecosystem with modules that bake reliability into the way they were built

CI/CD Integrations

Integrate automated rollbacks to minimize the risk of faulty changes.

Reliability



Principles
Support
Architecture

Config Templates

Extend config templates to include features that enable application reliability.

Runtime Packages

Extend package ecosystem with modules that bake reliability into the way they were built

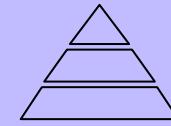
CI/CD Integrations

Integrate automated rollbacks to minimize the risk of faulty changes.

Incident Management Training

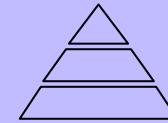
Build trust by providing documentation that aids with incident management.

Putting it all together

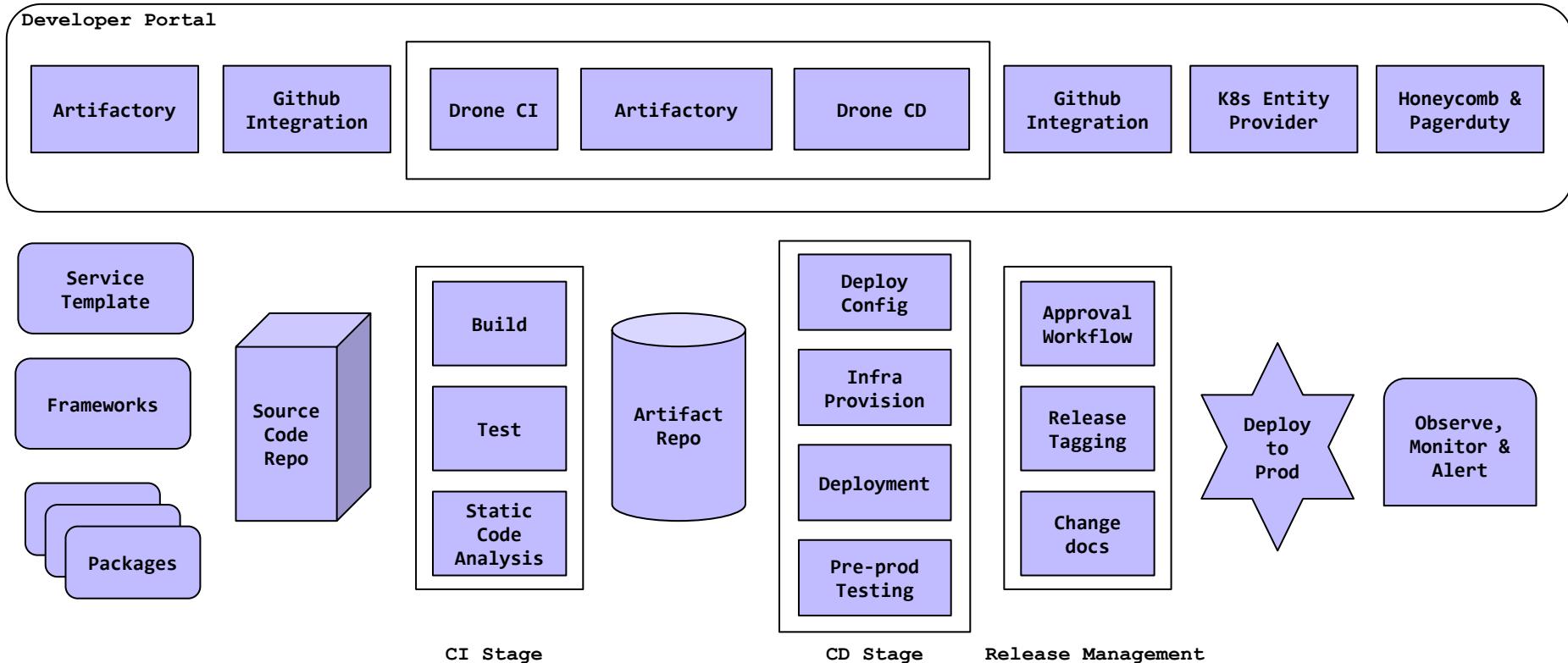


**Practices
Support
Architecture**

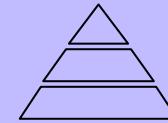
Lifecycle of a Service



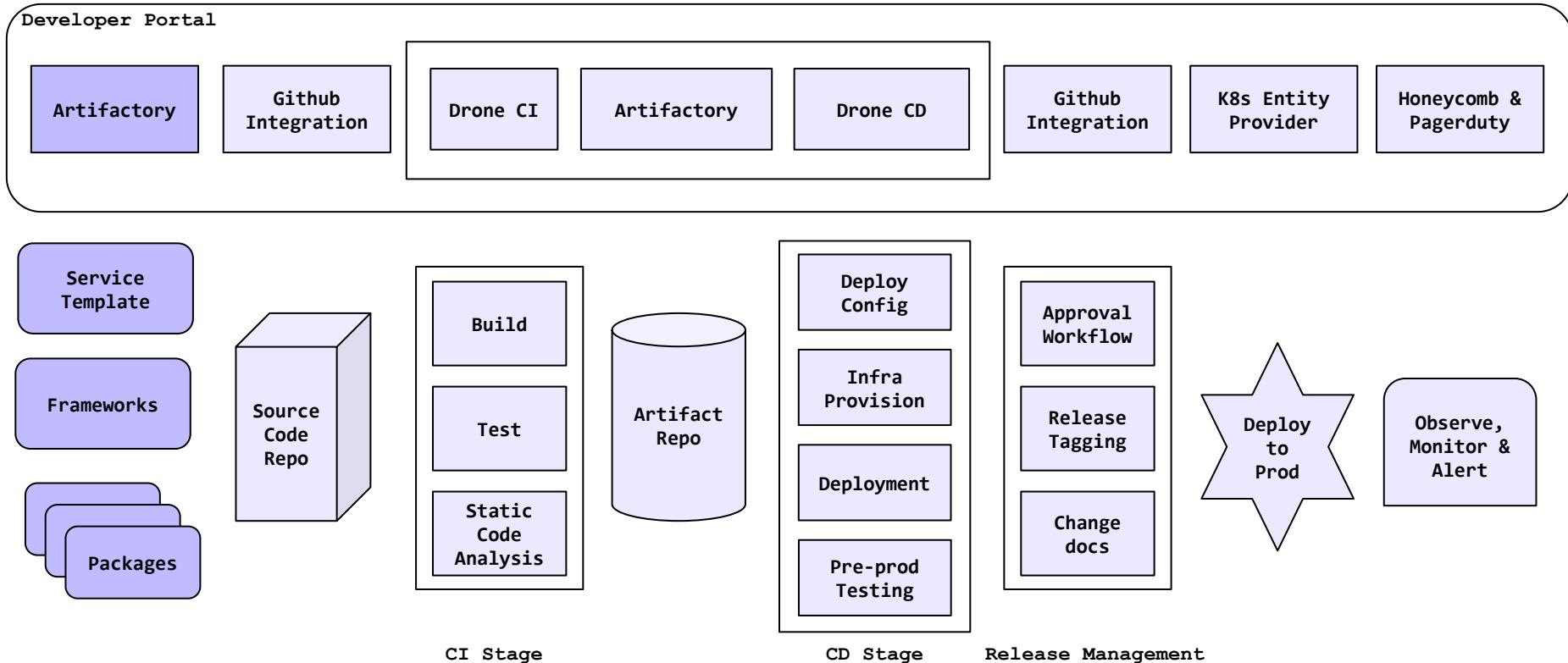
Principles
Support
Architecture



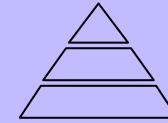
Lifecycle of a Service



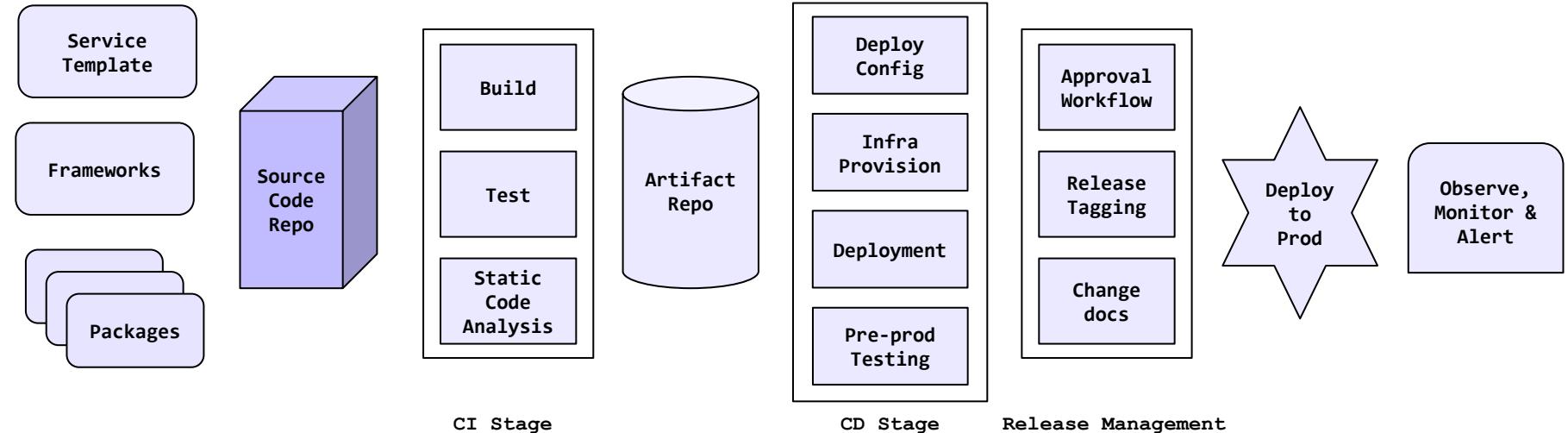
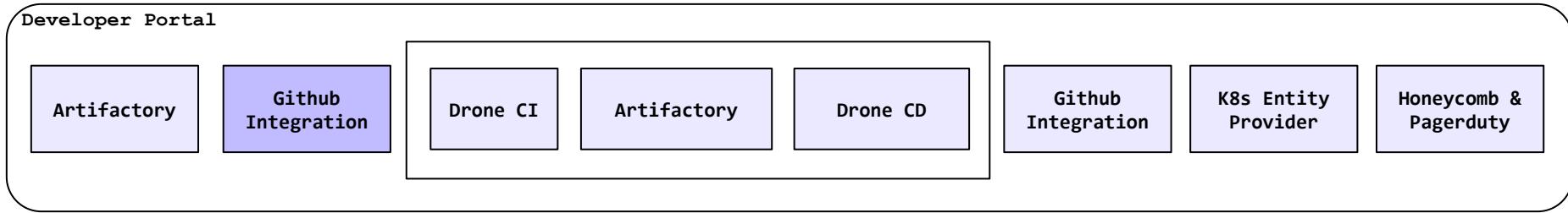
Principles
Support
Architecture



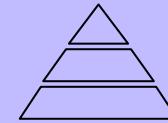
Lifecycle of a Service



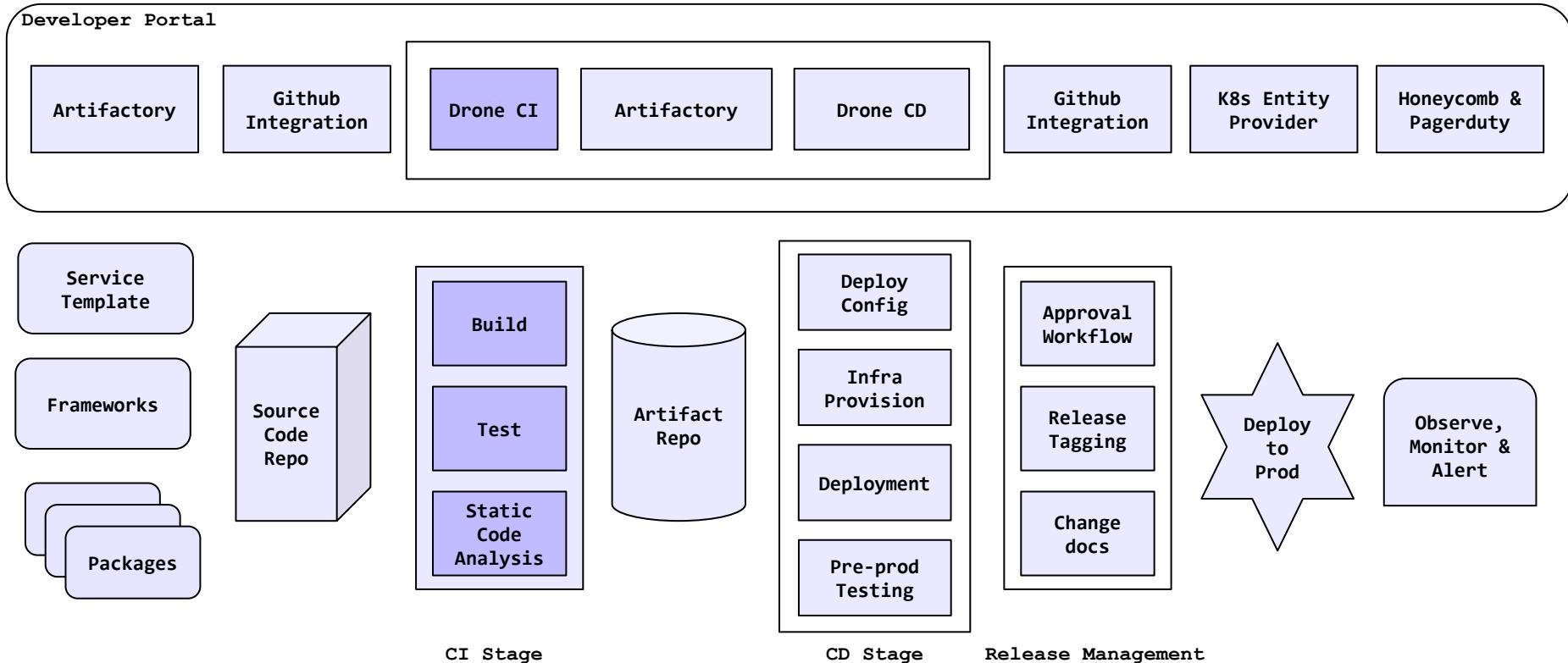
Principles
Support
Architecture



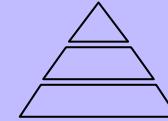
Lifecycle of a Service



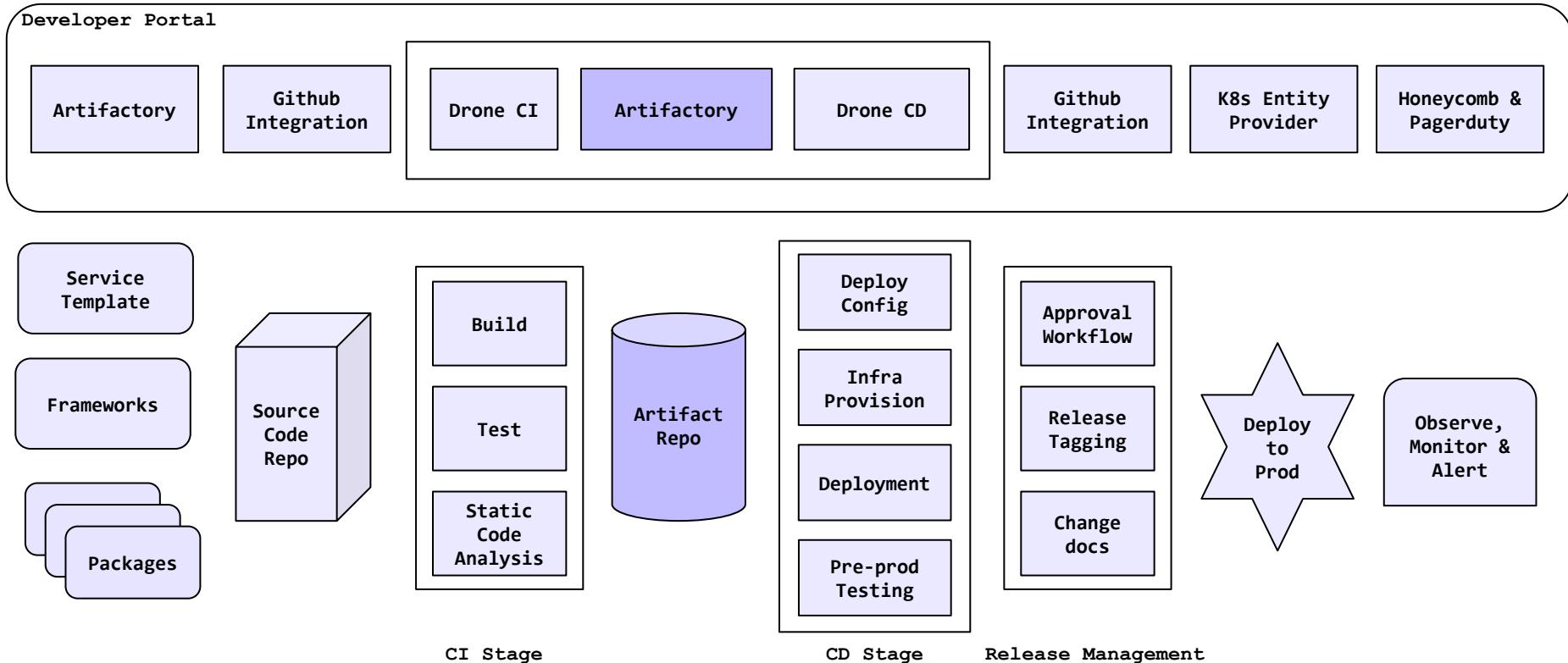
Principles
Support
Architecture



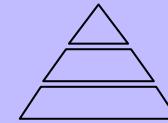
Lifecycle of a Service



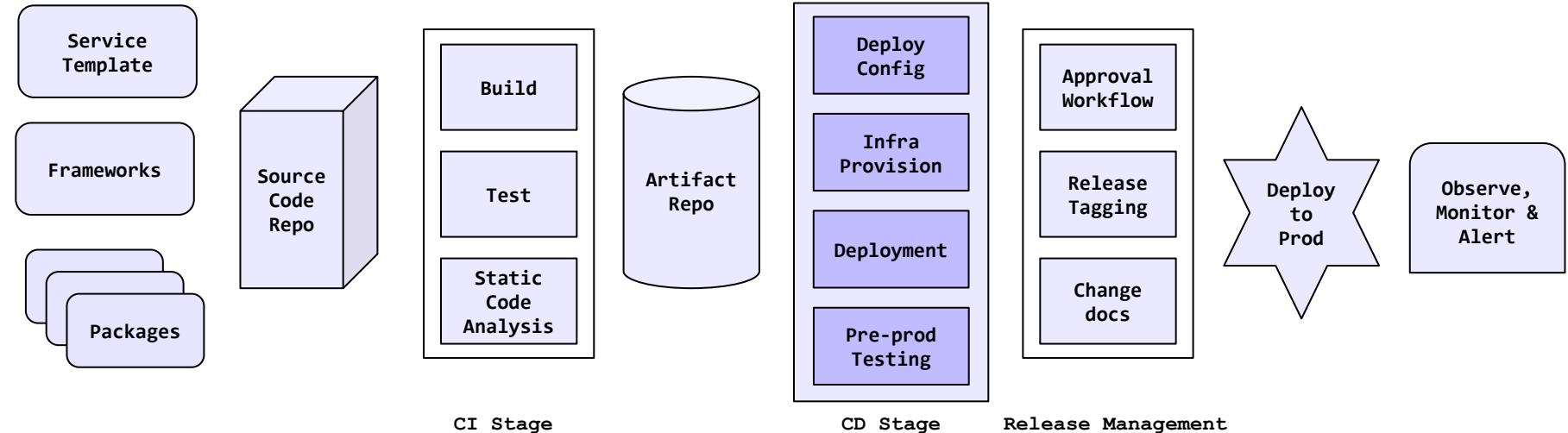
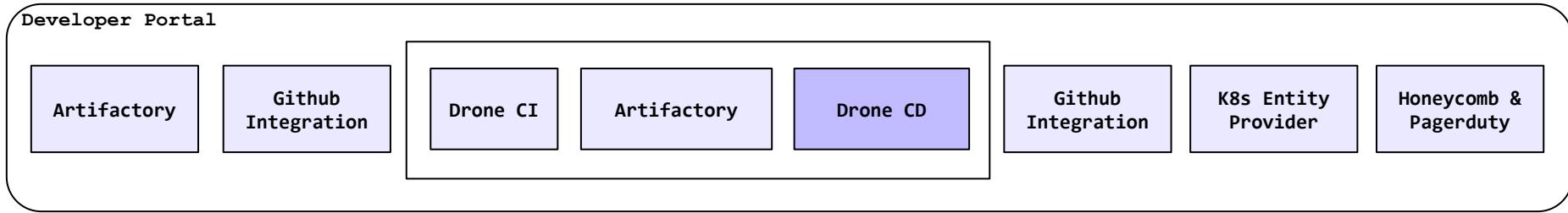
Principles
Support
Architecture



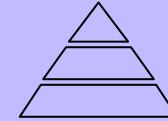
Lifecycle of a Service



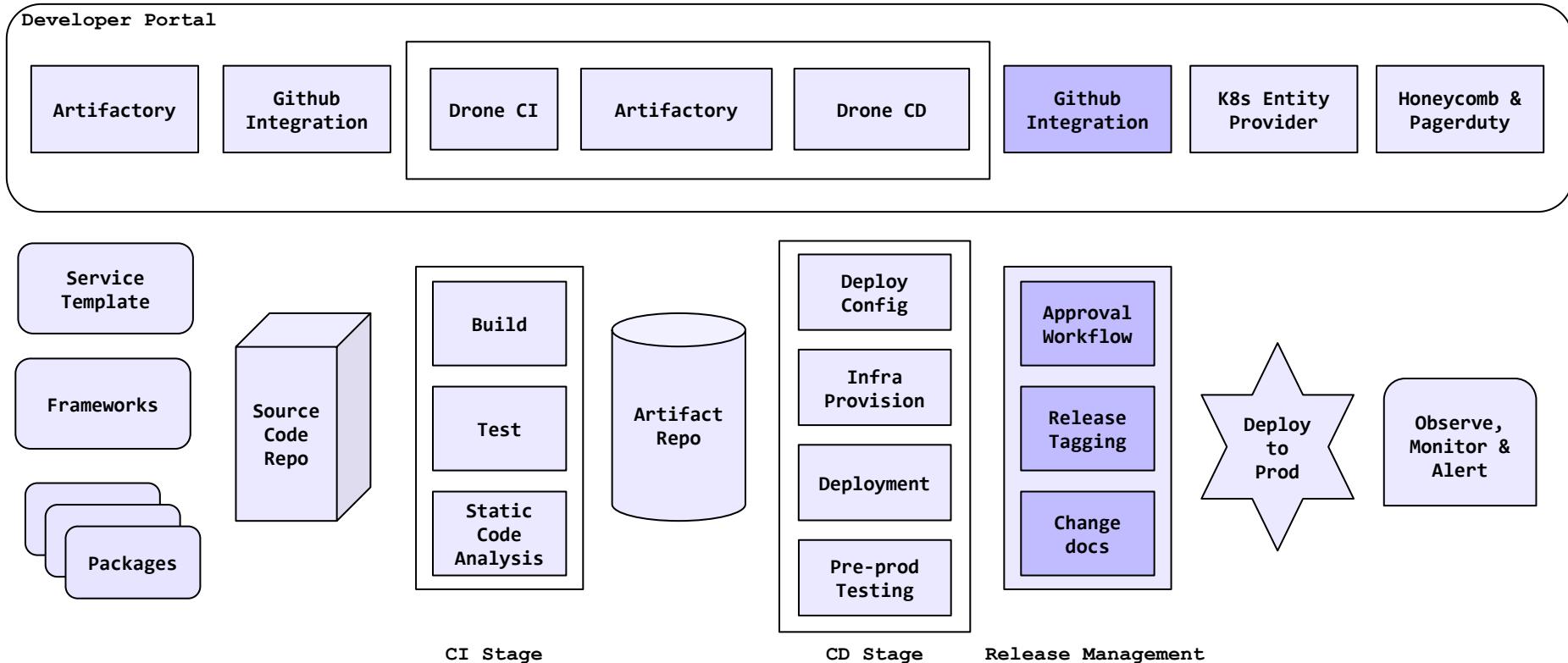
Principles
Support
Architecture



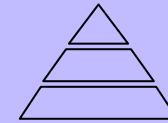
Lifecycle of a Service



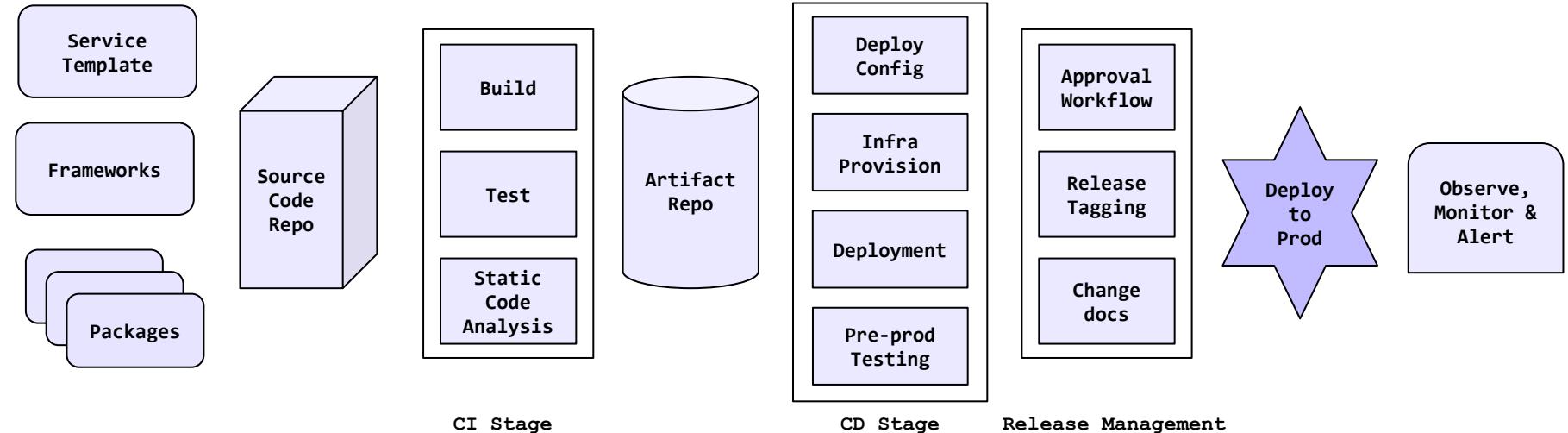
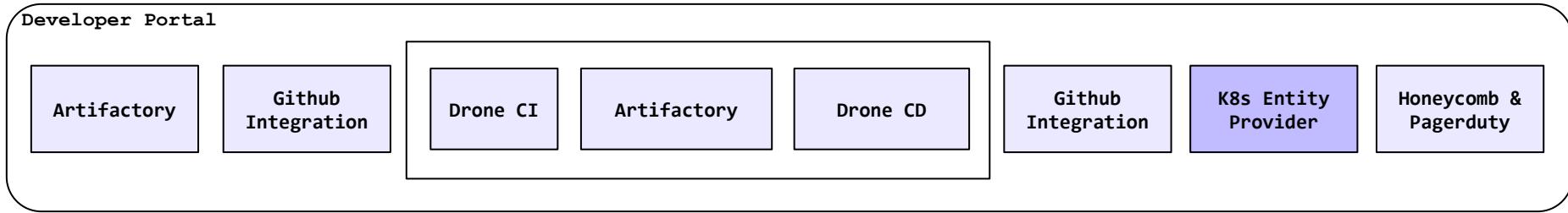
Principles
Support
Architecture



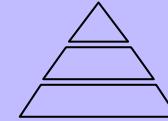
Lifecycle of a Service



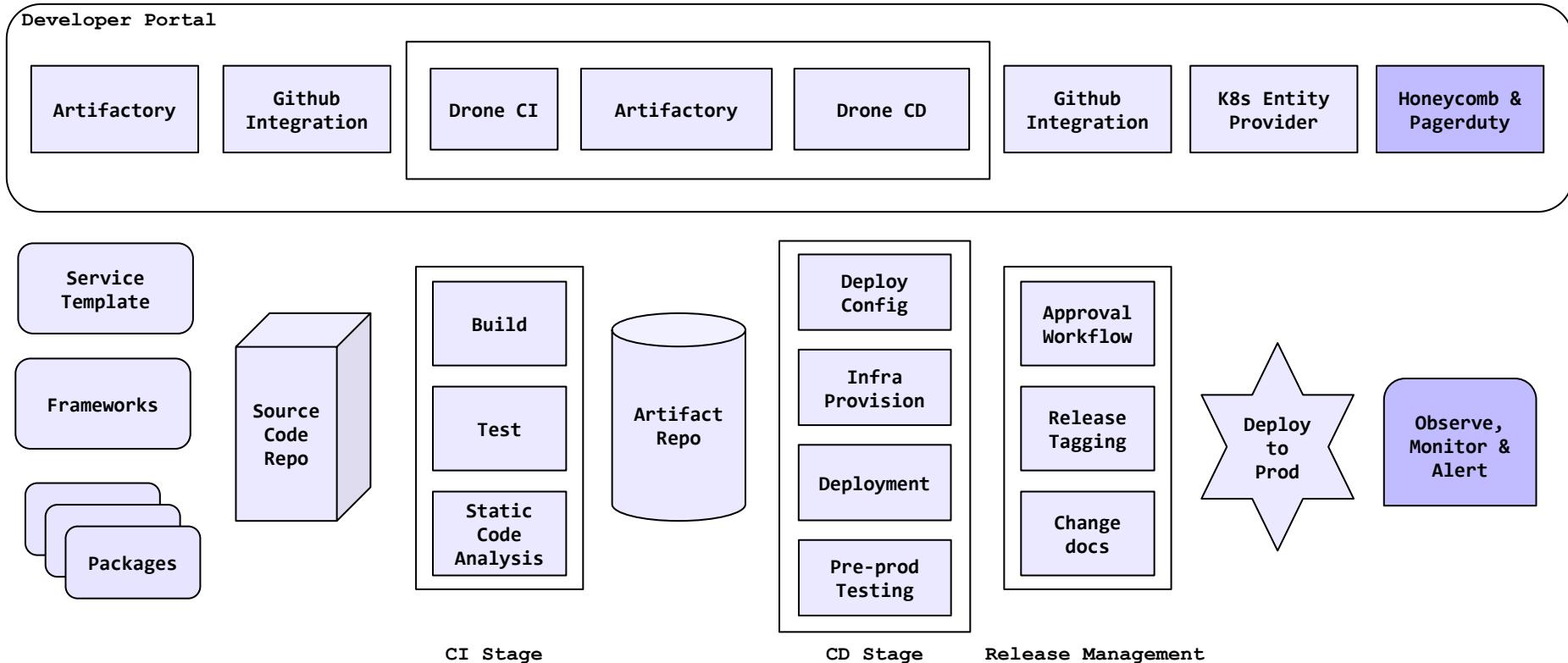
Principles
Support
Architecture



Lifecycle of a Service



Principles
Support
Architecture



Driving Sustainability with Platform Engineering



Lesley Cordero, @clesleycode
The New York Times, Staff Engineer