Introduction to DevOps & DevOps Philosophy

Maaike van Putten
Software developer & trainer
www.biltup.com

Overview

Session 1

Introduction to DevOps & DevOps Philosophy

Session 2

Introduction to CI/CD (Continuous Integration / Continuous Delivery)

Introduction round

- 💼 Current role, background and ambitions
- 🎨 Hobbies / interests / life outside of work
- 👩‍💻 What do you hope to learn?

Introduction to DevOps & DevOps Philosophy

Learning objectives

O

Define the work of the **operations team** in the traditional sense, understand its tasks and the role in the software development lifecycle.

P

Discuss the **philosophy** beyond DevOps, emphasizing collaboration, automation, continuous improvement, and high efficiency in development and operations teams.

Schedule

👋

Intro + theory

Introduction round + what is the operations team

b

Exercise

Manual deployments

D

Debrief + theory

Debrief exercise + challenges

b

Exercise

Thought experiment

q

DevOps theory

Terms and if enough time small exercises + mini quiz

Operations team

- Responsible for maintaining IT infrastructure
- Handles server setup, deployment and maintenance
- Manages network configurations and security
- Works separately from development teams
- Reactive approach to issues

Deploying software

- M Receive code from developers
- J Manually configure servers
- i Install dependencies and packages
- b Deploy code to production environment

Exercise

Manually deploy a basic application with a frontend and backend to experience the work of the operations team first-hand.

Debrief - Steps taken

- Server setup and configuration
- Installed dependencies manually
- Configured environment variables
- Deployed frontend and backend code
- Started application services

Challenges

- Time-consuming and error-prone
- Lack of standardization
- Minimal collaboration with developers
- Difficulty in reproducing environments

Exercise

Thought experiment: scaling up

Debrief

d Scaling up manually makes it very complex and time-consuming

Imaging having to manage all this

r Can you feel the headache yet?

What is DevOps?

- DevOps = Development + Operations
- A cultural movement, not just tools
- Unifies development and operation teams
- Focuses on collaboration and communication
- Enhances efficiency and quality

DevOps principles

- Collaboration
- Shared goals and responsibilities
- Automation
- Continuous Integration/Continuous Delivery (CI/CD
- Infrastructure as Code (IaC)
- Monitoring and logging
- Continuous feedback

Benefits of the software lifecycle

- Faster time to market
- m Improved quality
- X Enhanced collaboration and breaking down silos
- B Increased reliability
- h Automation reduces manual tasks
- R Continuous improvement

Philosophy behind DevOps

- n People over processes and tools
- B Shared responsibility and ownership
- b Lean and agile principles
- P Embracing failure as learning
- R Customer-centric focus
- D Culture of trust and transparency

Business value of DevOps

- Faster delivery meets market demands
- Optimized resources reduce expenses
- Improved quality, less errors due to automation
- Customer satisfaction
- Competitive advantage
- Risk mitigation and early detection of issues
- Scalability

O

Key Practices DevOps

Automation

- Automate repetitive tasks to free up human resources
- Implement CI/CD pipelines to streamline code integration and deployment
- Use automation tools such as Jenkins, GitHub Actions, GitLab CI/CD, CircleCi, Azure DevOps

Benefits of automation

- b Consistency and standardized processes
- M Faster delivery cycles
- k Reduced errors

Continuous improvement and feedback loops

- Measure performance
- Collect user feedback
- Iterative improvements to implement changes on feedback
- Tools like Prometheus, Grafana, ELK stack

Benefits of continuous improvement and feedback loops

R Higher quality

R Proactive issue resolution

e Informed decision-making

Collaboration between development and operations

X

Break down silos to foster cross-team interaction

B

Shared goals by aligning objectives and KPIs

O

Cross-functional teams that blend skills and expertise

Collaborative tools: Slack, Jira, Confluence

Benefits of collaboration between development and operations

S Improved communication

M Faster issue resolution

t Increased innovation

The three ways of DevOps

m

First way - flow (systems thinking)

- Optimize the entire system

R

Second way - feedback loops

- Amplify feedback for continuous improvement

A

Third way - continuous learning and experimentation

- Foster a culture of innovation and learning

CALMS framework

n **Culture:** collaboration and trust

h **Automation:** streamline processes

n **Lean:** eliminate waste

l **Measurement:** data-driven decisions

U **Sharing:** knowledge and success stories

DevOps maturity models

Levels of maturity:

- Initial (Ad Hoc)
- Managed
- Defined
- Measured
- Optimized

DORA metrics

- j Deployment frequency
- N Lead time for changes
- Mean time to recovery (MTTR)
- Change failure rate

Value stream mapping (VSM)

- What is VSM?
  - Visualizing the flow from idea to delivery
- Identifying bottlenecks
- Improving processes

DevOps toolchains and ecosystems

Tool categories:

- Planning and collaboration
- Source code management
- Continuous integration/delivery
- Monitoring and logging

Integrations and ecosystems

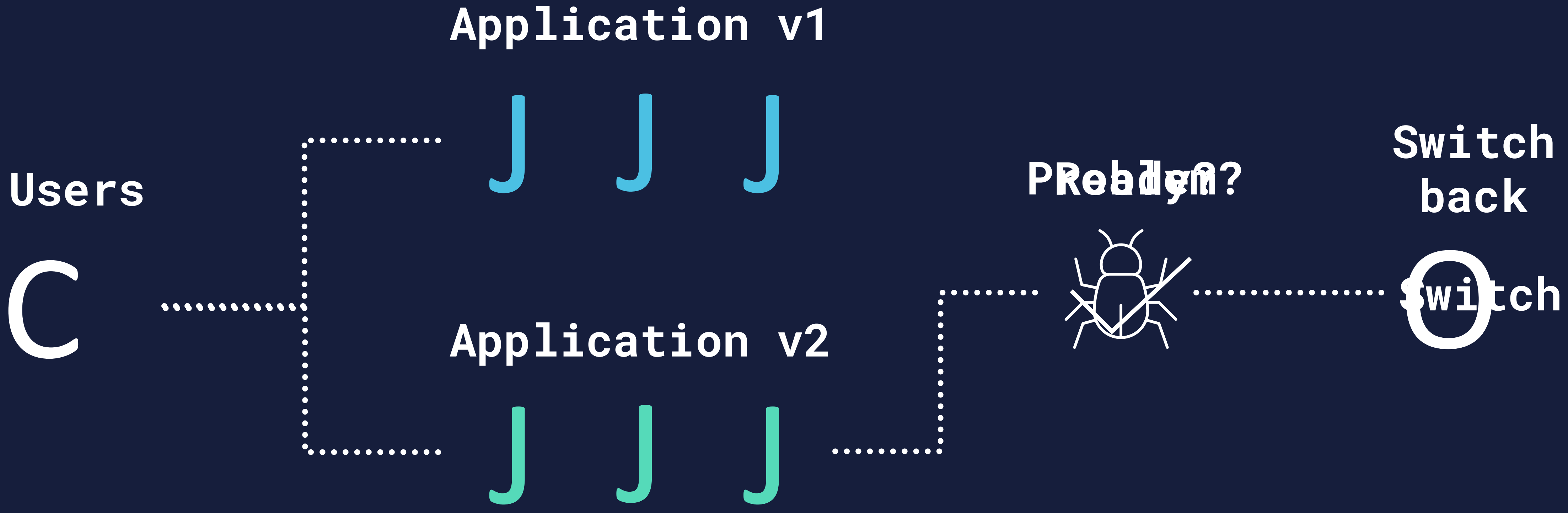
Exercise

Thought experiment: incorporating DevOps

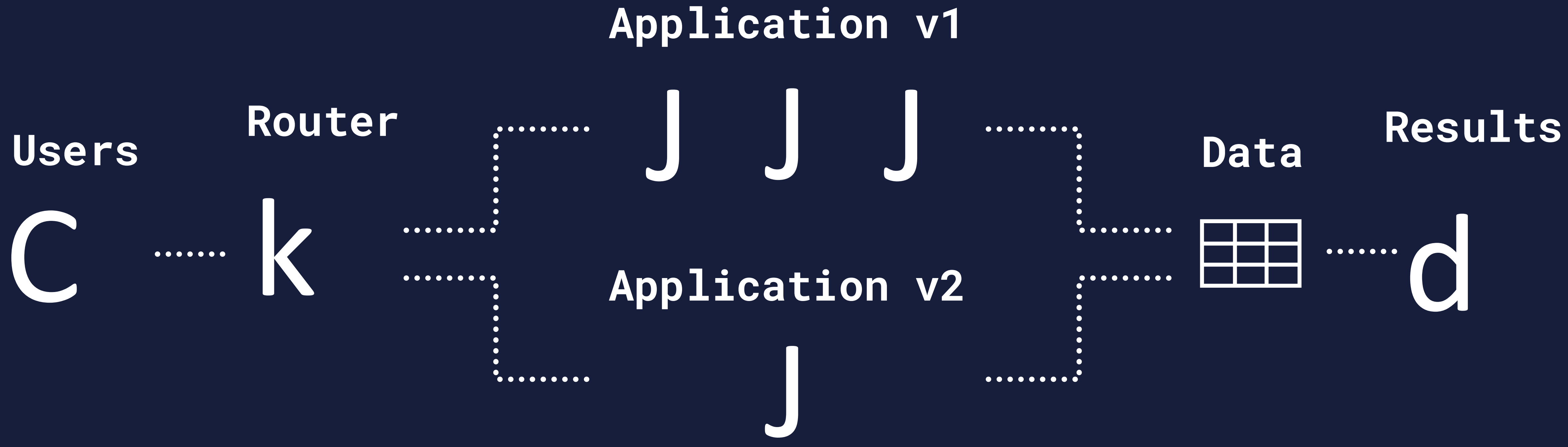
Deployment strategies

- E Blue/green deployment
- Canary releases
- Y Rolling updates
- U Feature toggles

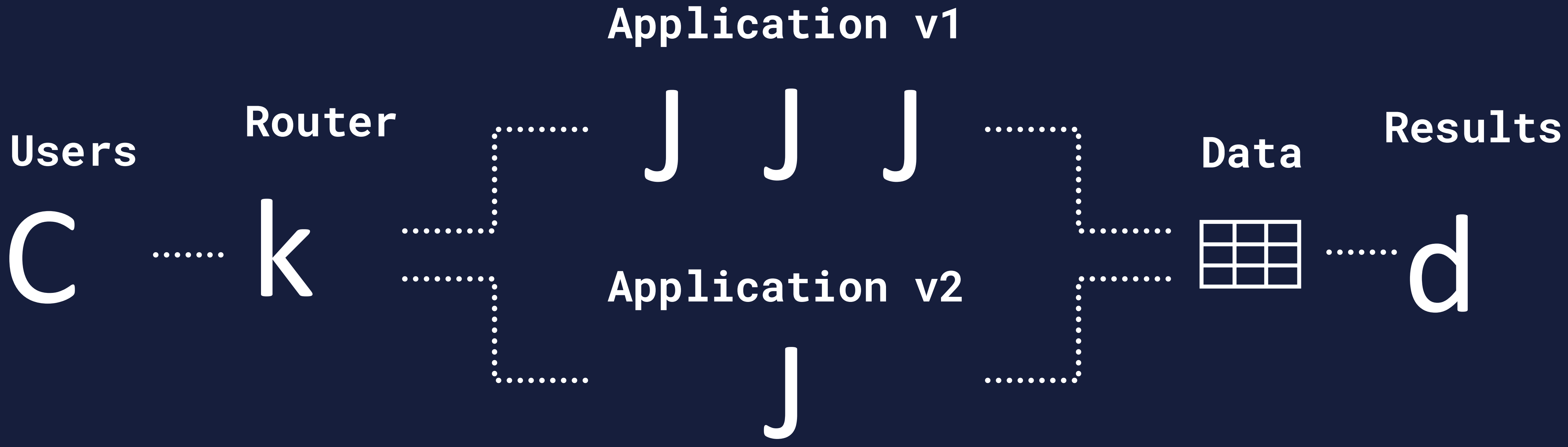
Blue/green deployment



Canary releases



Rolling update



Users

Router

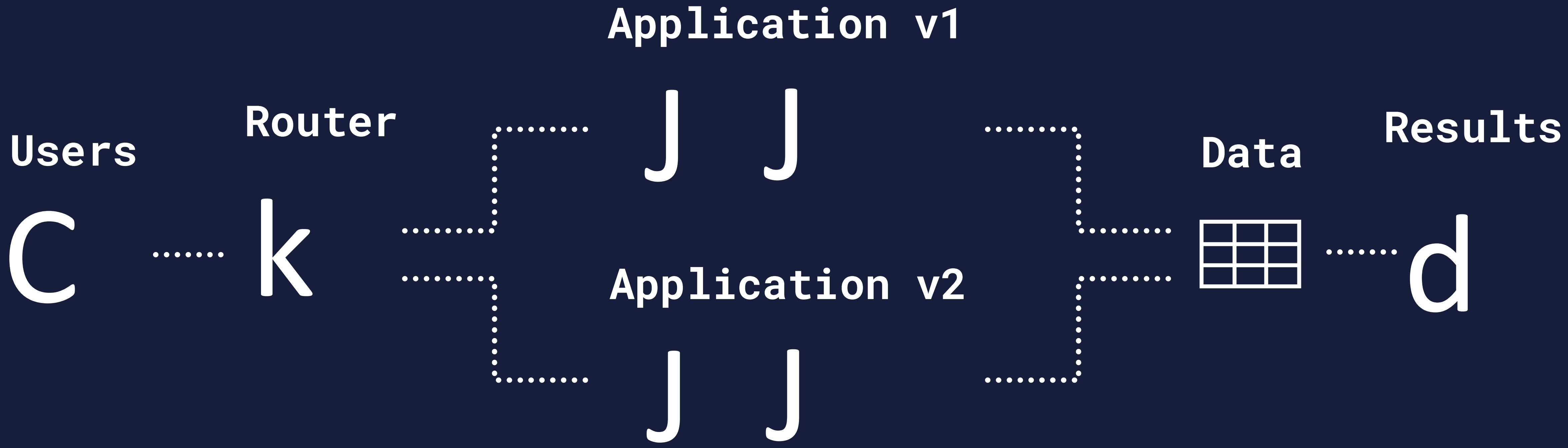
Application v1

Application v2

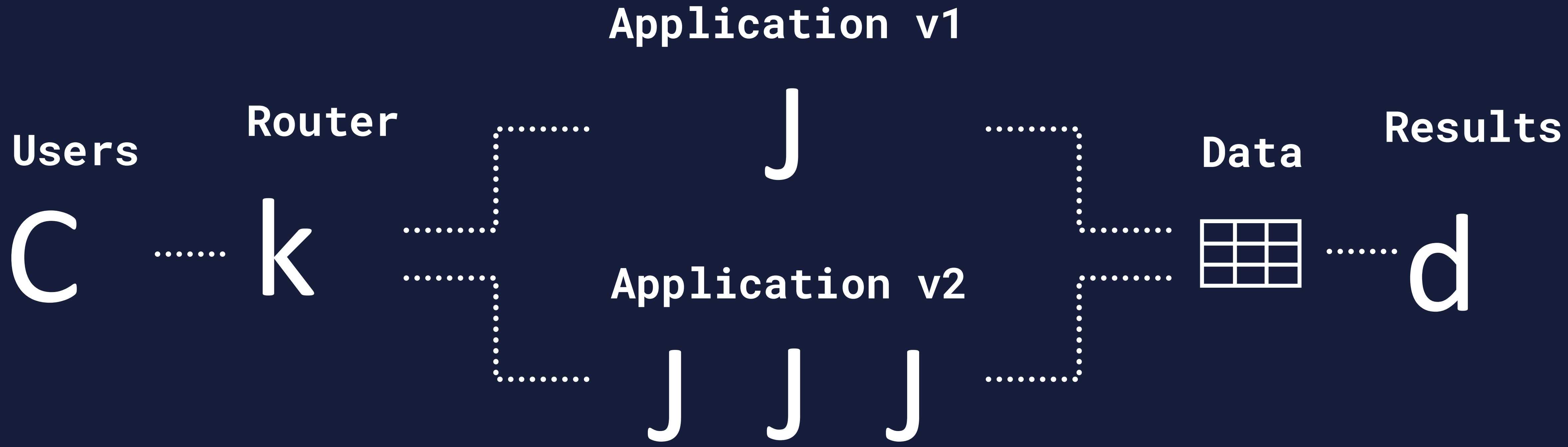
Data

Results

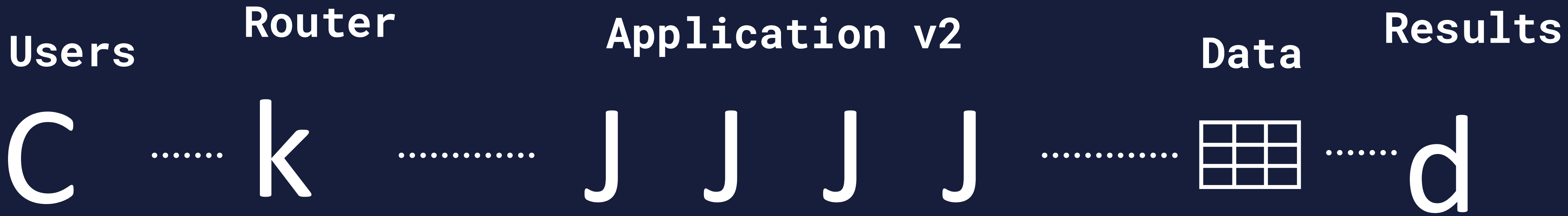
Rolling update



Rolling update



Rolling update



Feature toggles

- Let you adjust how a system works without needing to rewrite any code
- Allows features on or off as needed

- Useful for :
  - Testing new features on a specific user group
  - Quickly turning off a feature that causes issues

Exercise

Choosing a deployment strategy for our application

IaC and immutable infrastructure

IaC: Infrastructure as code

- Managing infrastructure with code

Immutable infrastructure

- No changes after deployment

Tools:

- Terraform
- Ansible
- CloudFormation

GitOps

- Git as the single source of truth

Principles:

- Declarative descriptions
- Automated deployments

Tools:

- Flux
- Argo CD

Site reliability engineering (SRE)

- Applying software engineering to operations

Principles:

- Reliability
- Scalability
- Efficiency

Key concepts:

- Service Level Objectives (SLOs)
- Error Budgets

Security in DevOps: DevSecOps

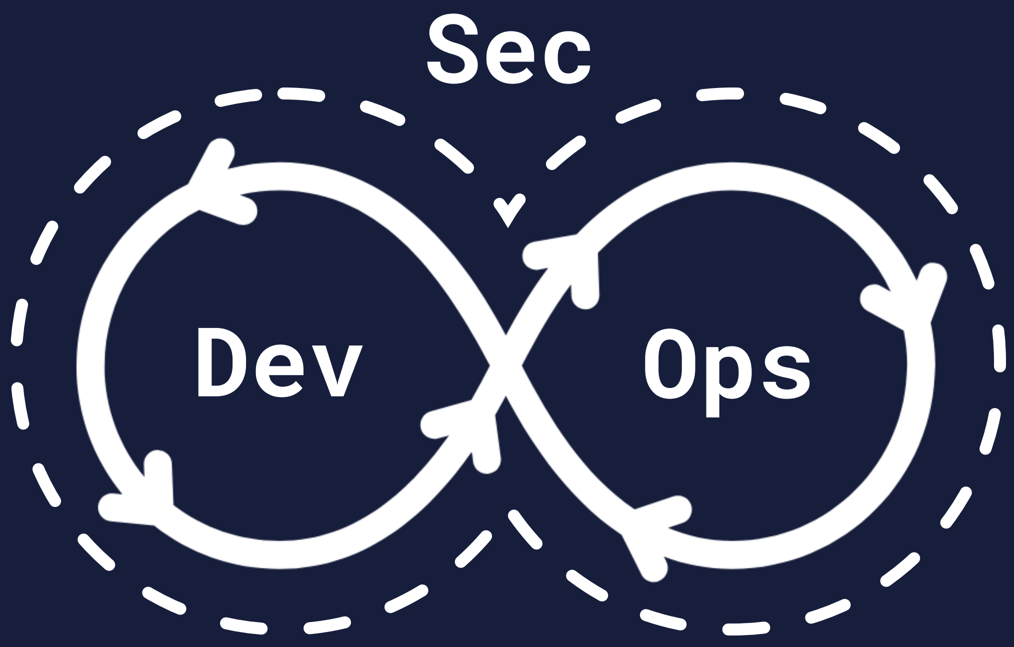
- Integrating security into DevOps

Principles:

- Shift-left security
- Automation of security checks

Tools:

- Snyk
- OWASP ZAP



What is the primary goal of DevOps in the software development lifecycle?

{A} To eliminate the need for testing by automating deployments

{B} To separate development and operations teams to increase specialization

{C} To outsource operations tasks to third-party vendors

{D} To unify development and operations teams for enhanced collaboration and efficiency

What is the primary goal of DevOps in the software development lifecycle?

{A} To eliminate the need for testing by automating deployments

{B} To separate development and operations teams to increase specialization

{C} To outsource operations tasks to third-party vendors

{D} **To unify development and operations teams for enhanced collaboration and efficiency**

Which of the following is NOT one of the Three Ways of DevOps?

{A} Rigorous compliance enforcement

{B} Amplify feedback loops

{C} Continuous learning and experimentation

{D} Flow (systems thinking)

Which of the following is NOT one of the Three Ways of DevOps?

{A} **Rigorous compliance enforcement**

{B} Amplify feedback loops

{C} Continuous learning and experimentation

{D} Flow (systems thinking)

In the CALMS framework, what does the 'L' stand for?

{A} Lean

{B} Leadership

{C} Learning

{D} Lifecycle

In the CALMS framework, what does the 'L' stand for?

[✓ {A} Lean]

{B} Leadership

{C} Learning

{D} Lifecycle

What is the purpose of Value Stream Mapping (VSM) in a DevOps context?

{A} To create a hierarchy of team responsibilities

{B} To design the user interface for applications

{C} To visualize and analyze the flow of work to identify bottlenecks

{D} To map network infrastructure for security purposes

What is the purpose of Value Stream Mapping (VSM) in a DevOps context?

{A} To create a hierarchy of team responsibilities

{B} To design the user interface for applications

{C} **To visualize and analyze the flow of work to identify bottlenecks**

{D} To map network infrastructure for security purposes

Which deployment strategy involves running two identical production environments where one is live and the other is on standby, allowing for quick rollbacks?

{A} Rolling update

{B} Blue/Green deployment

{C} Canary release

{D} Feature toggle deployment

Which deployment strategy involves running two identical production environments where one is live and the other is on standby, allowing for quick rollbacks?

{A} Rolling update

[✓ {B} **Blue/Green deployment**]

{C} Canary release

{D} Feature toggle deployment

Next up:

Introduction to CI/CD

Questions or suggestions?

maaikejvp@gmail.com

See you tomorrow!