Communication & Cooperation

Unity®

2022

Jami Kousa

- Graduated with Masters from University of Helsinki in 2019
- E-sports professional
- Teacher at the University of Helsinki for ~4 years

- 1 year at Unity as a Software Engineer in the Services Foundation team.
- Full-stack, DevOps, Infrastructure

Let’s get started

- → What is Unity?
- → What is the Unity Gaming Services?
- → What is the Services Foundation?
- → How important communication actually is?


Unity®

Unity and real-time 3D

→ **GAMES**

More than 50% of all games are made with Unity

→ **ARCHITECTURE**

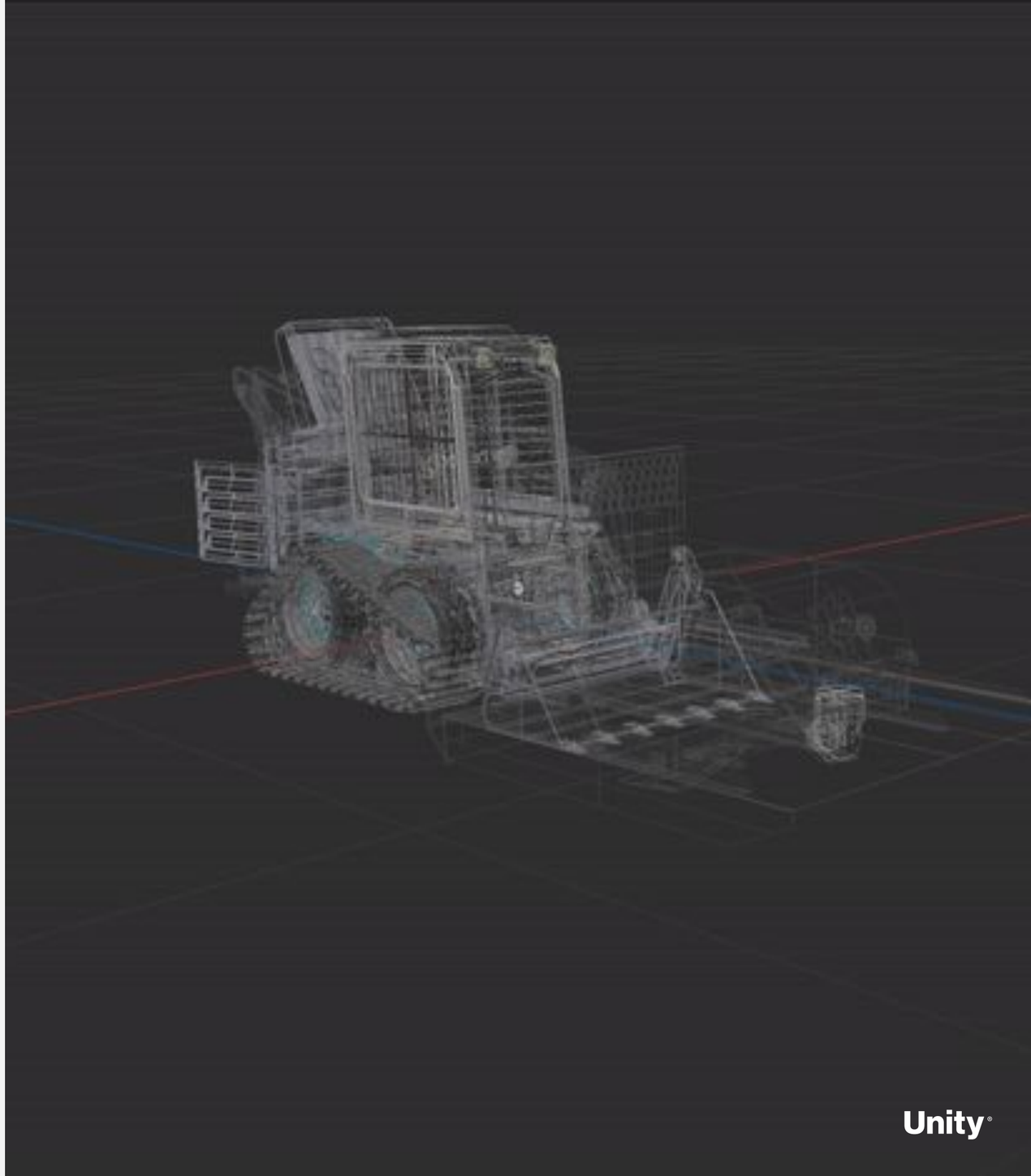
Tools are used in Building Information Modeling across the industry

→ **AUTOMOTIVE**

Industry leaders such as Toyota use Unity as means to improve design, engineering and training.

→ **FILM**

Instead of waiting hours or days for rendering – you remove all barriers to artistic experimentation.



2K

Software
Engineers & Developers

1,5M

Monthly Creators

2B

Monthly Users

Unity Gaming Services

A selection of tools

→ **CLOUD CODE**
Run game logic in the cloud.

→ **CLOUD DIAGNOSTICS**
Identify and resolve the bugs in games.

→ **CLOUD SAVE**
Storing game data to the cloud.

→ **ECONOMY**
Design and plan game economy.

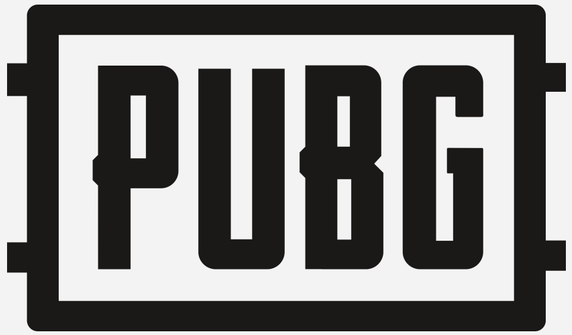
→ **MULTIPLAY**
Game Server Hosting for multiplayer with dedicated game servers.

→ **RELAY**
Multiplayer without dedicated game servers.

→ **UNITY ANALYTICS**
An end-to-end data and analysis solution for games

→ **VIVOX**
In-game voice and text chat software

Games using our services



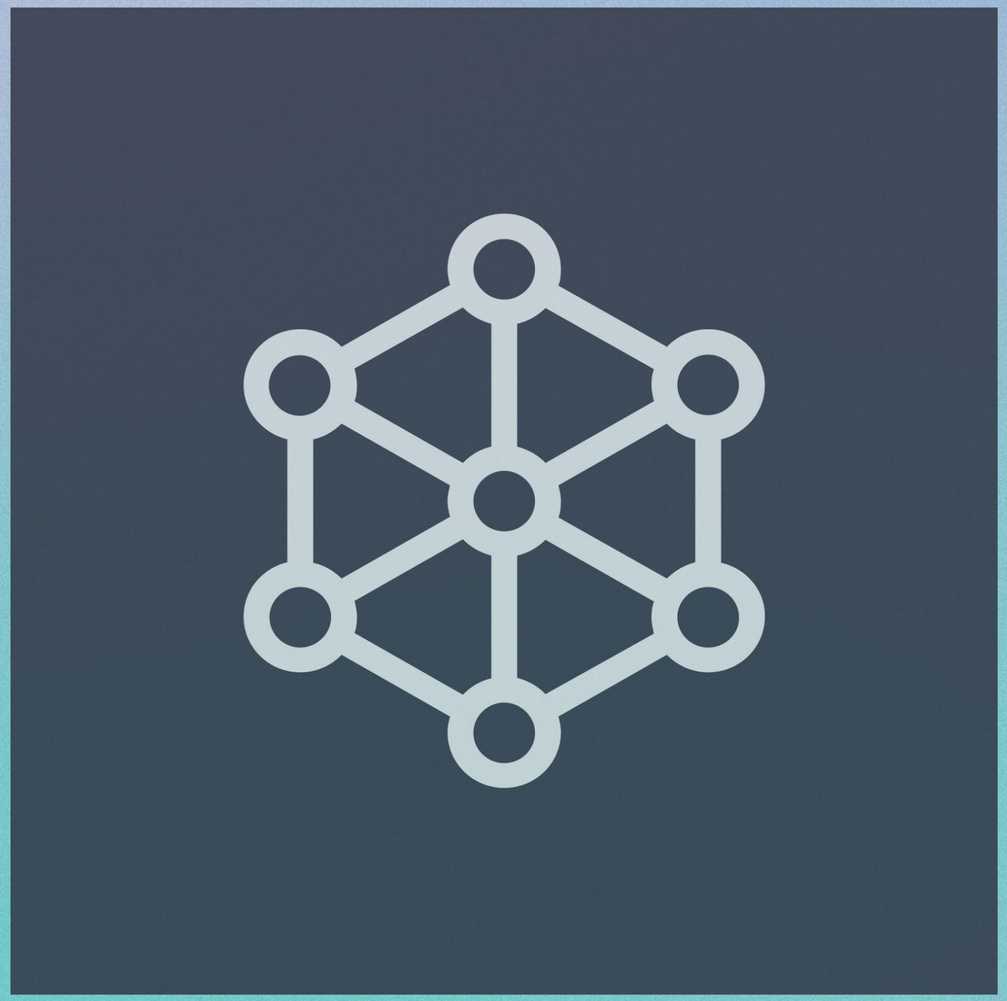
Services Foundation

Who are we?

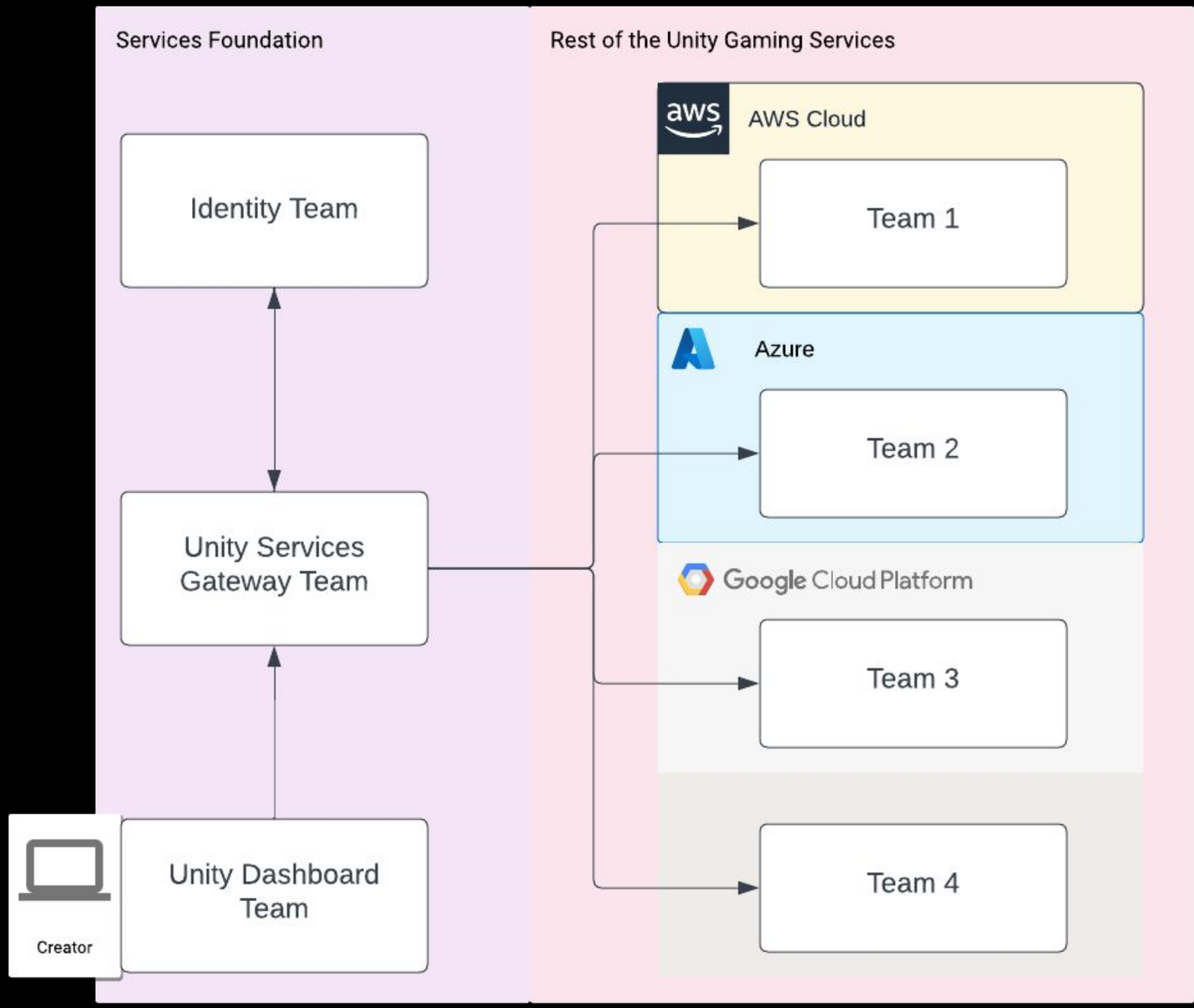
- A group split into many smaller teams all with distinctive responsibilities

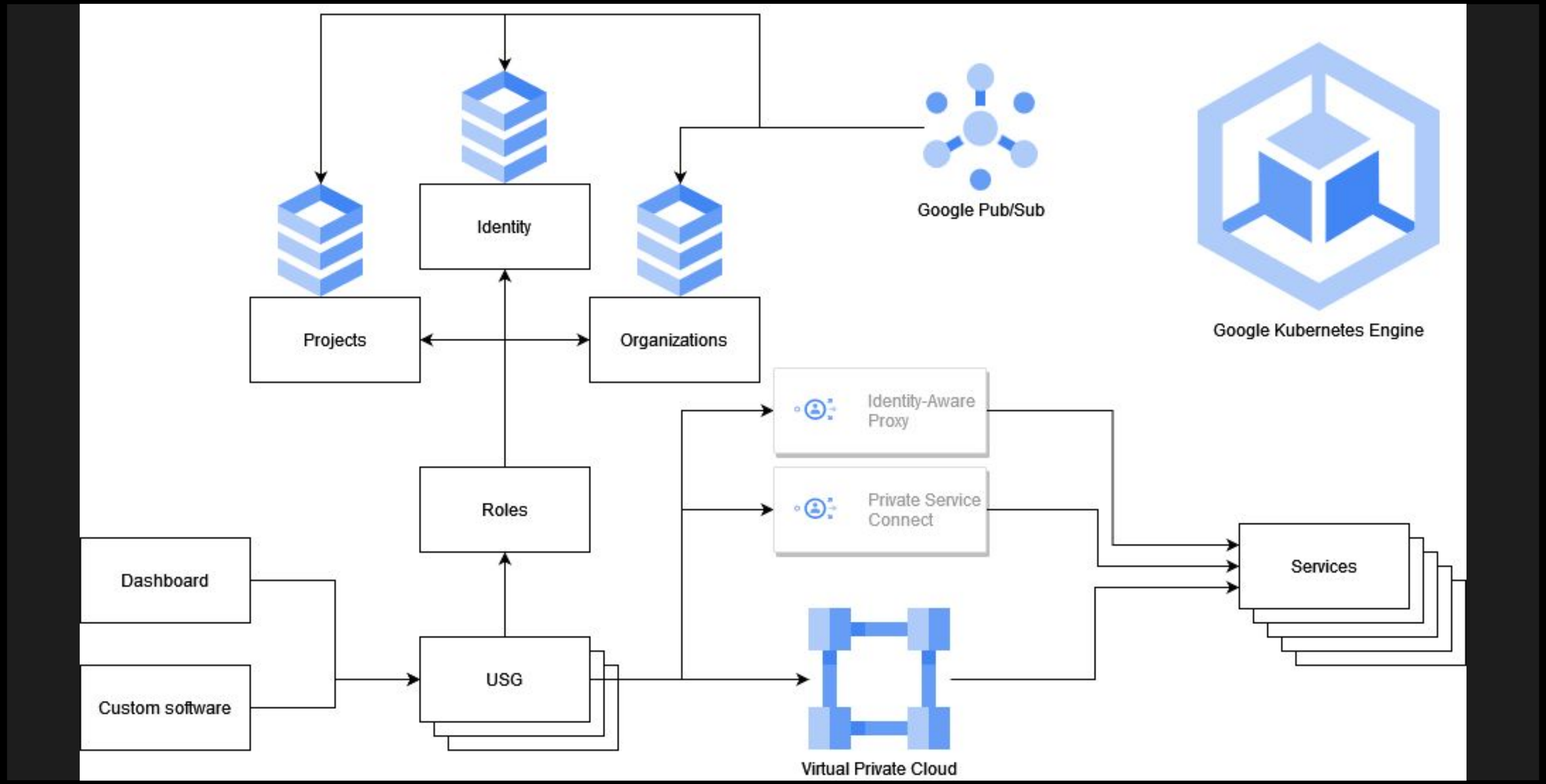
What do we do?

- Maintain a selection of products used by teams of Unity Gaming Services.

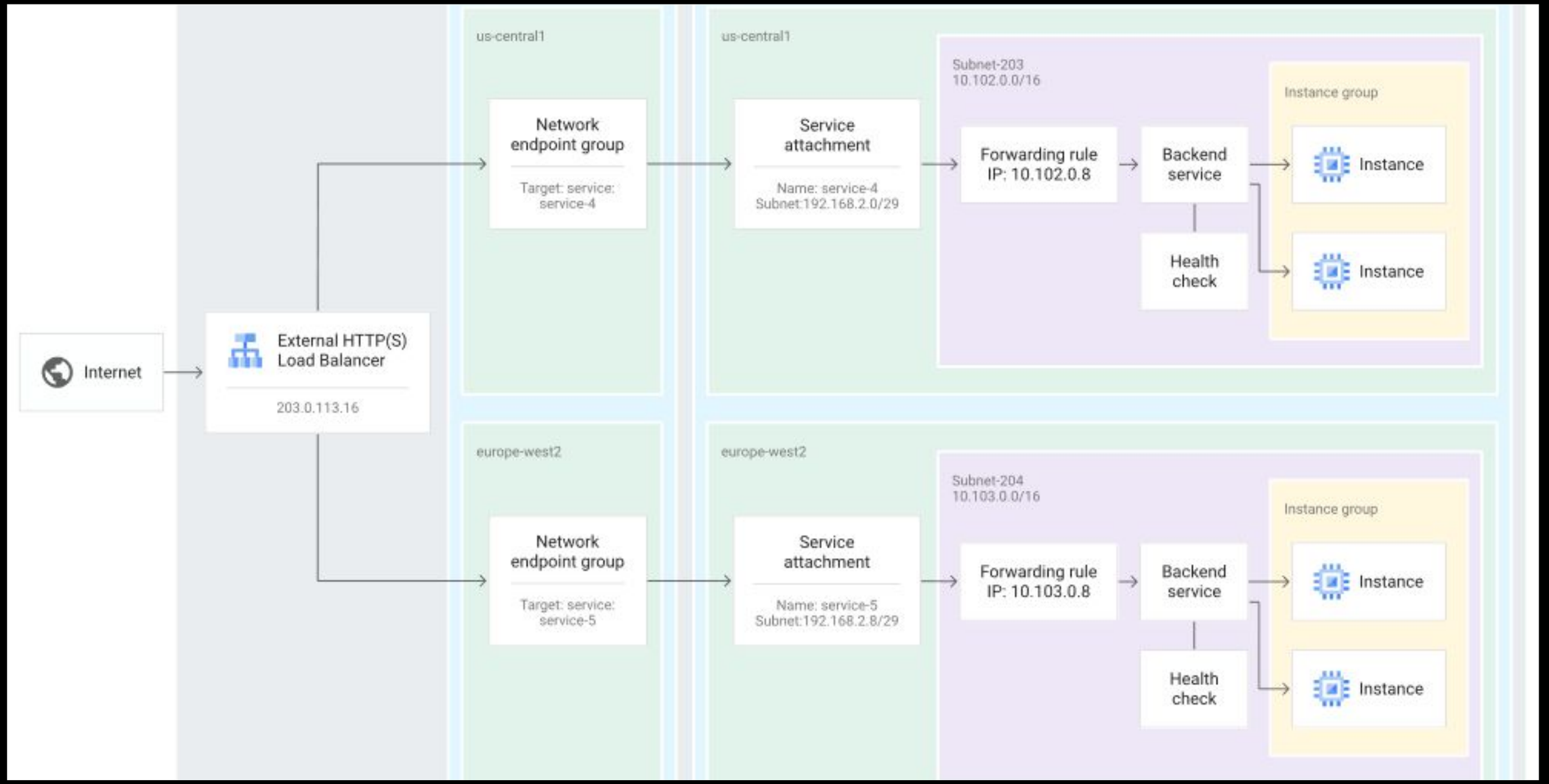


What do I do?


Services Foundation
Rest of the Unity Gaming Services
Identity Team
Unity Services Gateway Team
Unity Dashboard Team
Creator
aws
AWS Cloud
Team 1
Azure
Team 2
Google Cloud Platform
Team 3
Team 4

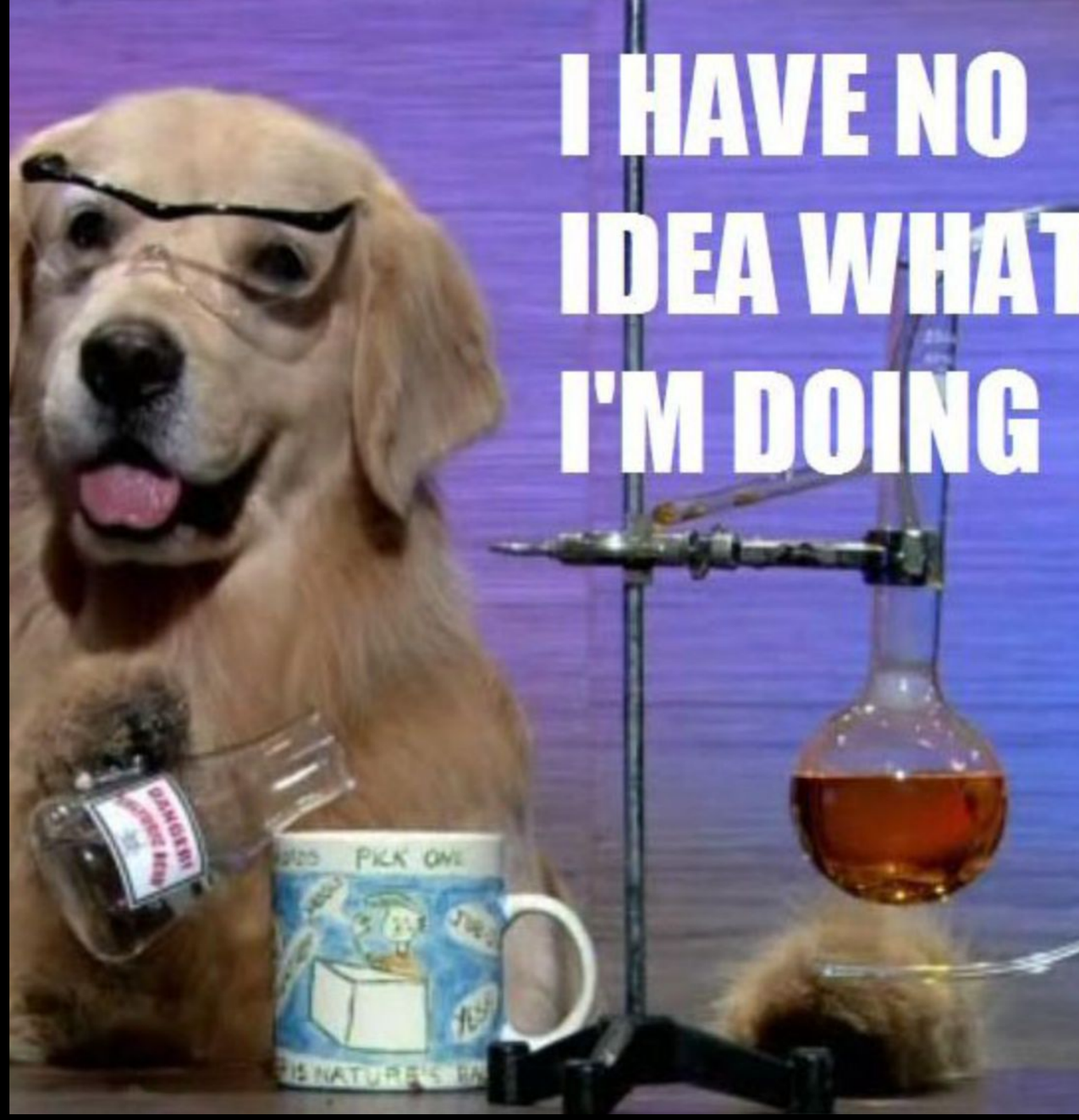


→ **A Closer Look** The complexity increases the closer we zoom in.



→ **Private Service Connect** To zoom into one of the integration methods and how it's used

What do I actually do?



The biggest challenge

Communication

Three steps

→ **BEFORE**

- How to not lose focus
- Spread responsibility

→ **DURING**

- Make sure you’re doing the right thing
- Spread responsibility

→ **AFTER**

- How to survive an incident

Documents

- PRODUCT REQUIREMENTS DOCUMENT
- TECHNICAL DESIGN DOCUMENT
- ARCHITECTURE DECISION RECORD

Product Requirements Document

- **WHAT**
  What is needed
- **WHY**
  Why it’s needed
- **END RESULT**
  How it should function
- **NON-TECHNICAL**
  Readable by non-developers

Technical Design Document

→ **HOW**
How the requirements can be met

→ **OPTIONS**
Offer optional solutions

→ **SHARING**
Team agrees on the solution

→ **VERY TECHNICAL**
Focus is on engineer audience

Architecture Decision Record

→ **WHY**
Contains the reasoning for the decision

→ **OPTIONS**
Shows optional solutions and why they were not chosen

→ **LOW-BARRIER**
Written to document decision process

→ **OPEN TO EVERYONE**
The document is stored for everyone to read

Architecture Decision Record

- → **AUTHOR & TEAM**
  Who made the document
- → **CONTEXT & WHY**
  The reason for the document and a description of the problem
- → **REQUIREMENTS**
  Functional & non-functional requirements outlined
- → **OPTIONS**
  What kind of options there are to solve the problem
- → **DECISION**
  Outlines the decision you would prefer from the options.
- → **IMPACT**
  What other consequences the decision has. These are things outside of the requirements.

Code Reviews

- → Trust you’re doing the right thing
- → Quality is maintained
- → Responsibility is shared

Making a good Pull Request

→ Think of the reviewer
→ Don’t make a PR you wouldn’t review
→
→ Make a PR early

Incidents

Incident process

ALERTS

There are many ways for incidents to start.

- Often it’s an automated alert which gets triggered in some service somewhere
- Sometimes it’s noticed by developers or creators

CONFIRM THE ISSUE

After you notice something is wrong, starts the evaluation whether it’s a problem or something minor.

- Sometimes alerts get triggered because of faulty code.
- Sometimes alerts get triggered by accident or because of a misconfigured alert.

INCIDENT CHANNEL

If the problem is confirmed we start an incident channel.

- Incident commander is assigned
- Pull in the people who would be the most useful
- Update service status on the status website.
- Nobody works after working hours so make sure you have someone ready to pick up the commander mantle.

“If it hurts, do it more often”

— Martin Fowler

To paint a picture

- **Over 12 000 slack channels**
  Slack channels are created to communicate with teams about important topics, such as incidents.

- **Over 24 000 emojis**

- **Ever evolving context**
  Leaving a paper trail is the way to do a feature and forget it the next day, and still communicate about it with hundreds of developers.

Thank you

Unity®

UNITY.COM