

Tame your microservices with Ortelius



Juan Peredo

The humble pizza

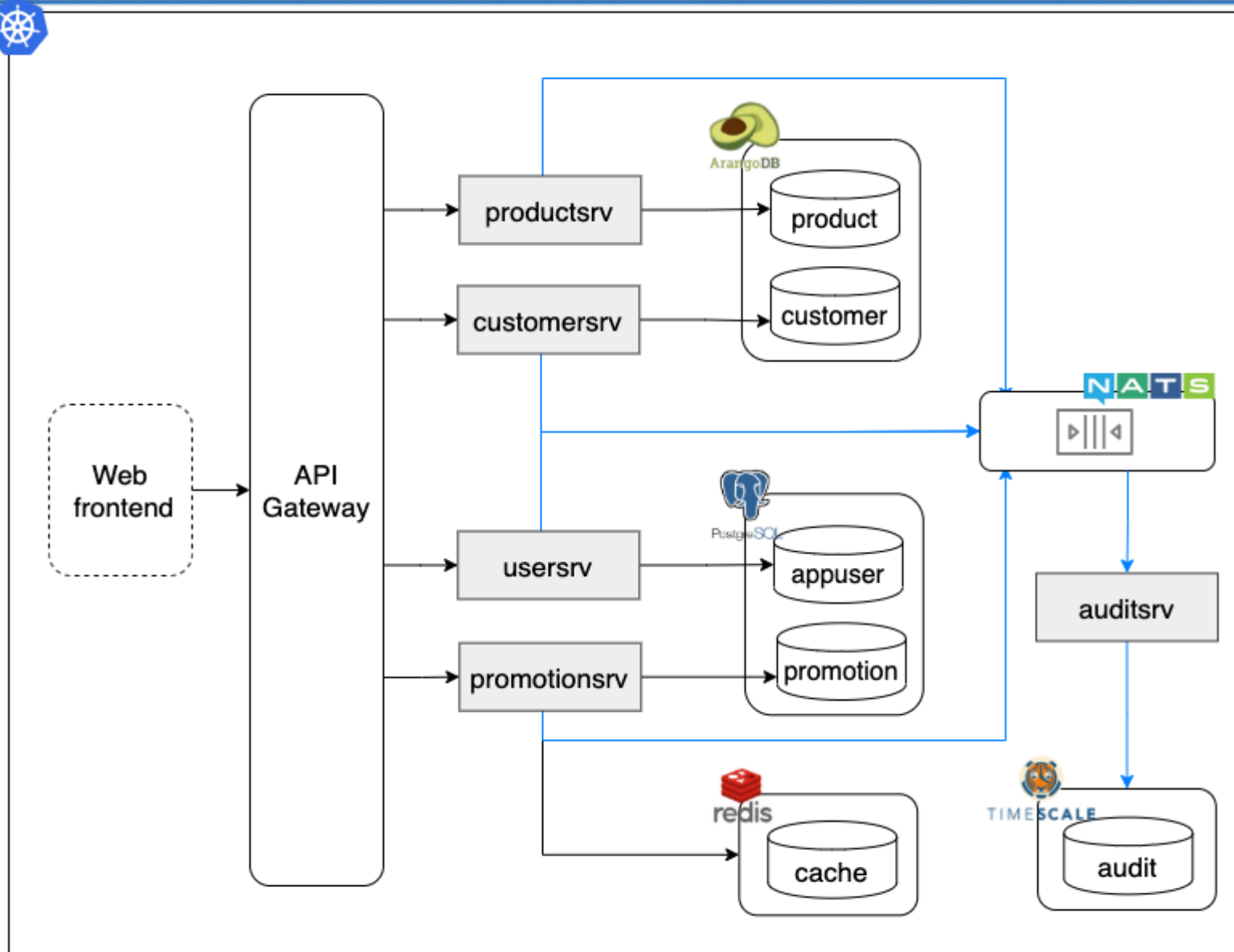




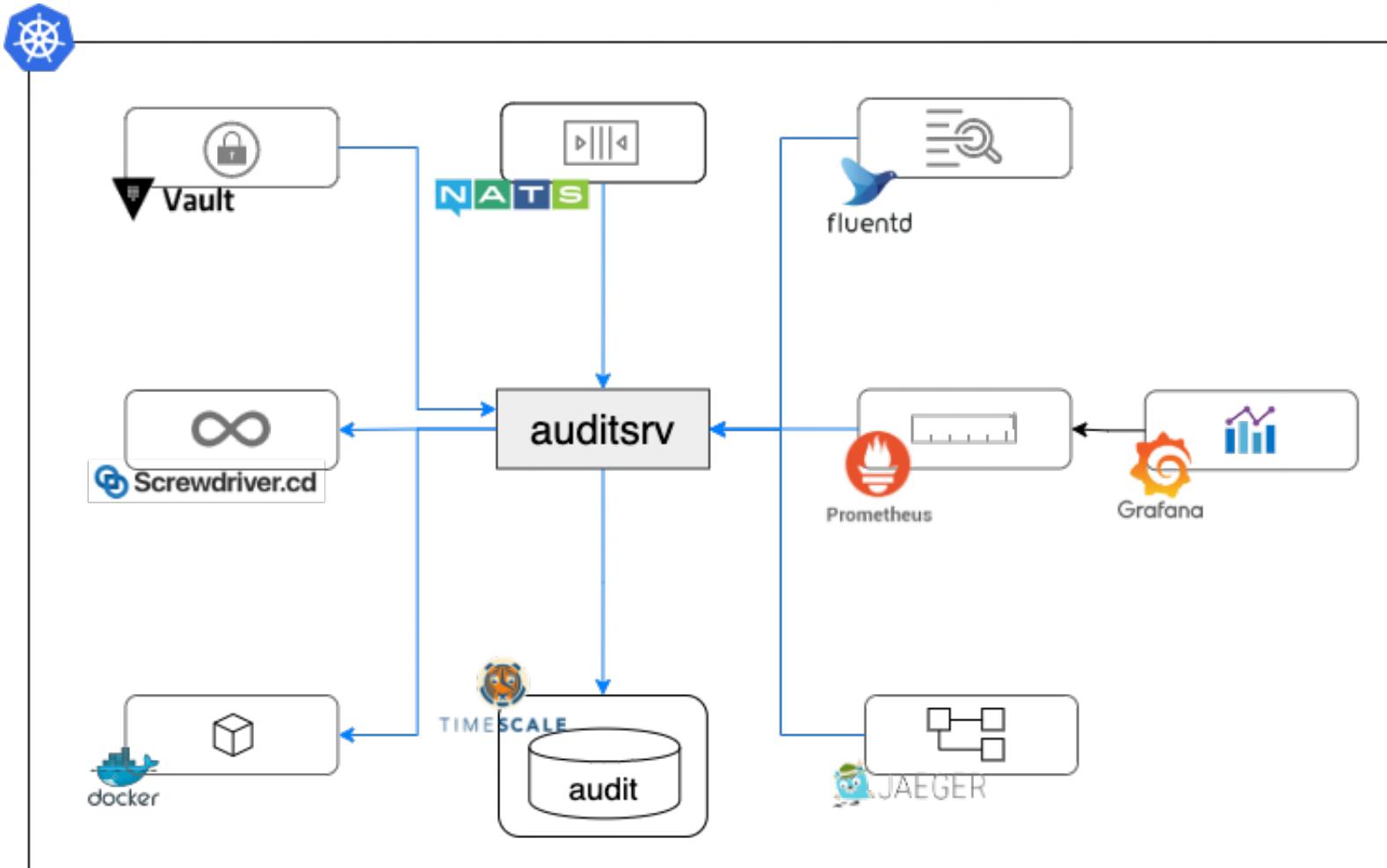
The ingredients

Secret to a good Pizza:

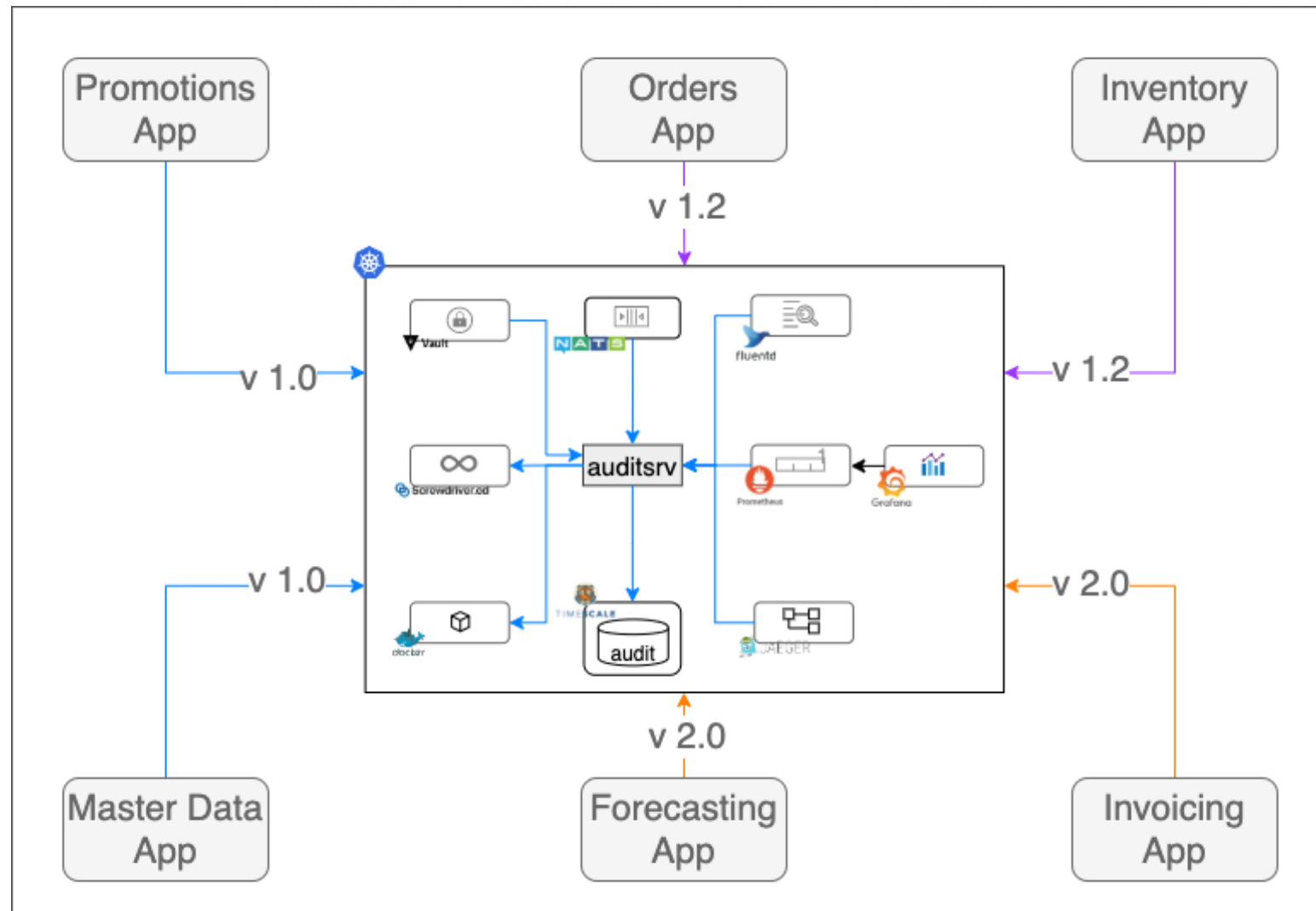
- Ingredient quality
- Ingredient management
- Ingredient mix



- Loose coupling of microservices
- Additional dependencies
- Potentially multiple tech stacks
- Independent deployments



- Service itself is loose coupling of multiple technologies



- Service used by multiple applications
- There could be multiple versions of a microservice active at any one time

What could go wrong?



- Modern apps have lots of moving parts
- Very hard to:
 - Keep track of the app components
 - Update individual components
 - Determine source of failure



Move from application management to microservice management



Tech consultant / architect / developer and everything in between.

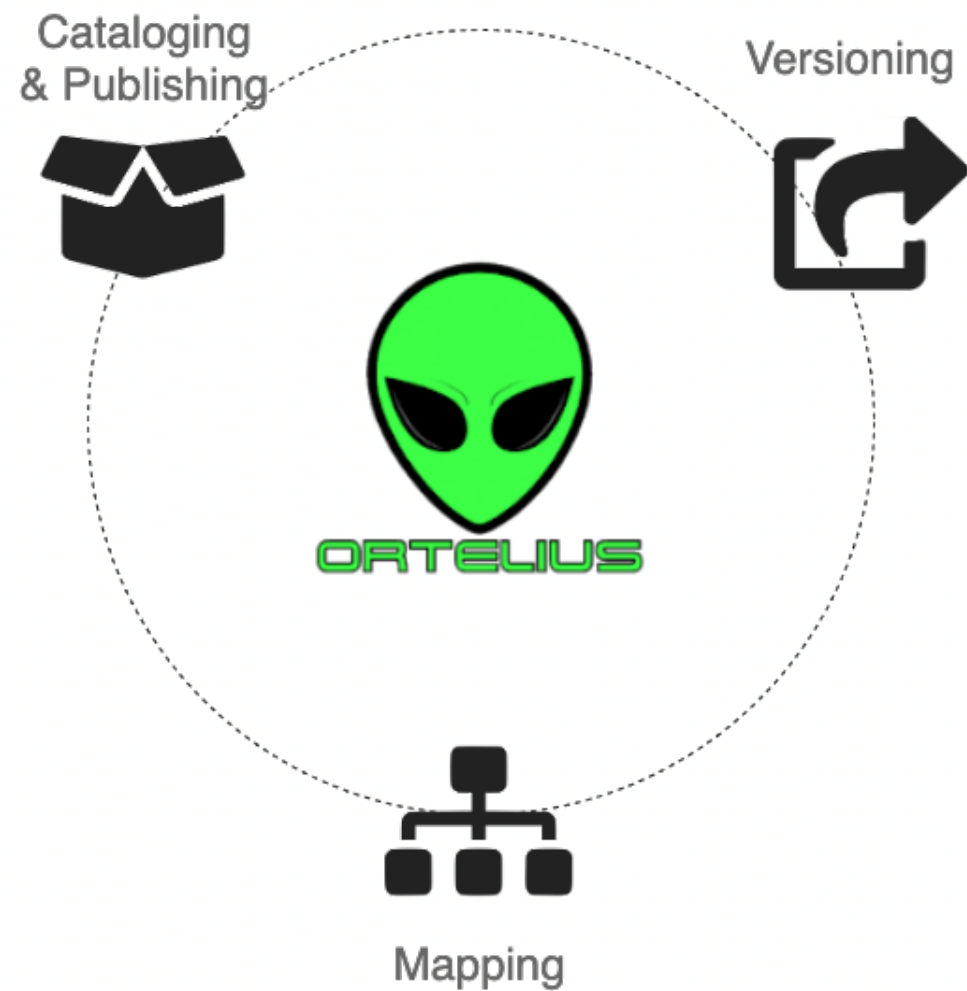
Github: <https://github.com/camba1>

Twitter: @JuanPeredoTech

Linkedin: <http://linkedin.com/in/juanperedotech>

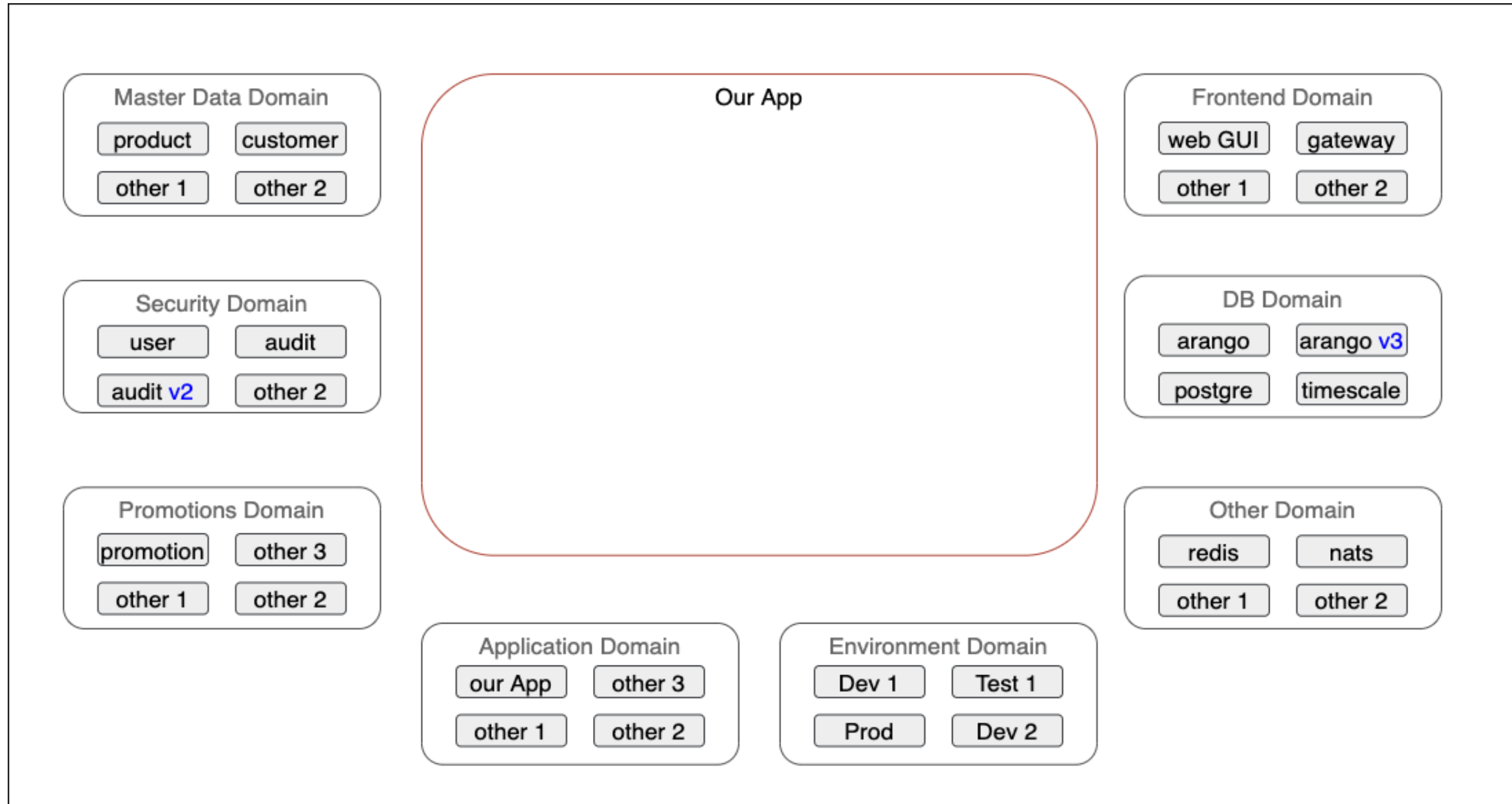
Application View	<ul style="list-style-type: none">• Which microservices are used ?• Which version of a microservice is used ?• What are the dependencies of each microservice ?
Microservice View	<ul style="list-style-type: none">• Which applications use the microservice?• What is the upgrade blast radius?• When to sunset a version?• What are the dependencies?• Which environment contains which microservice?

Crucial Information to manage hundreds of microservices

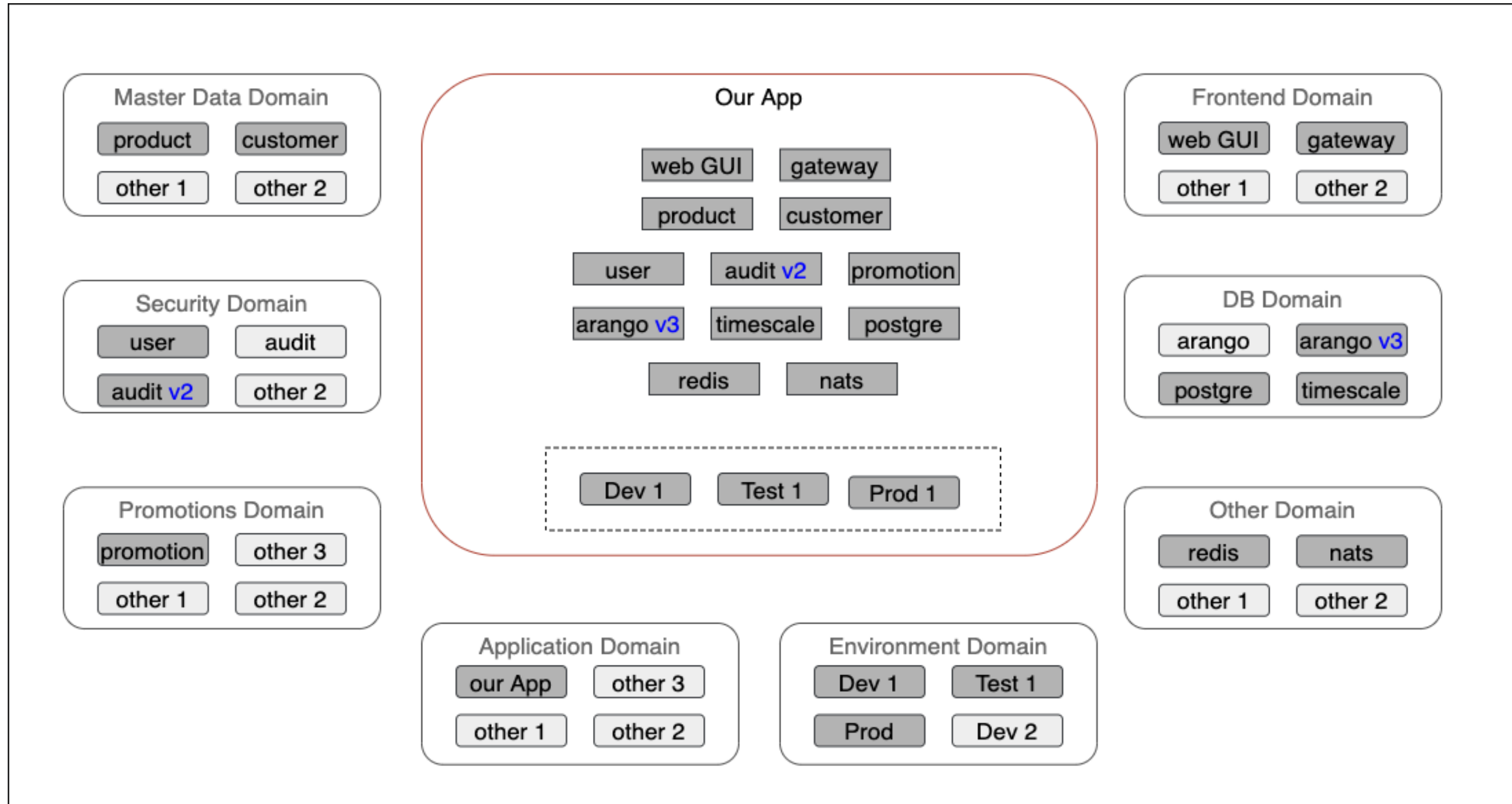


- Open source project
- Started on : December 2020
- Part of the CD foundation
- Project home: [Ortelius.io](https://ortelius.io)
- Documentation: docs.ortelius.io/guides/

Domain Catalog



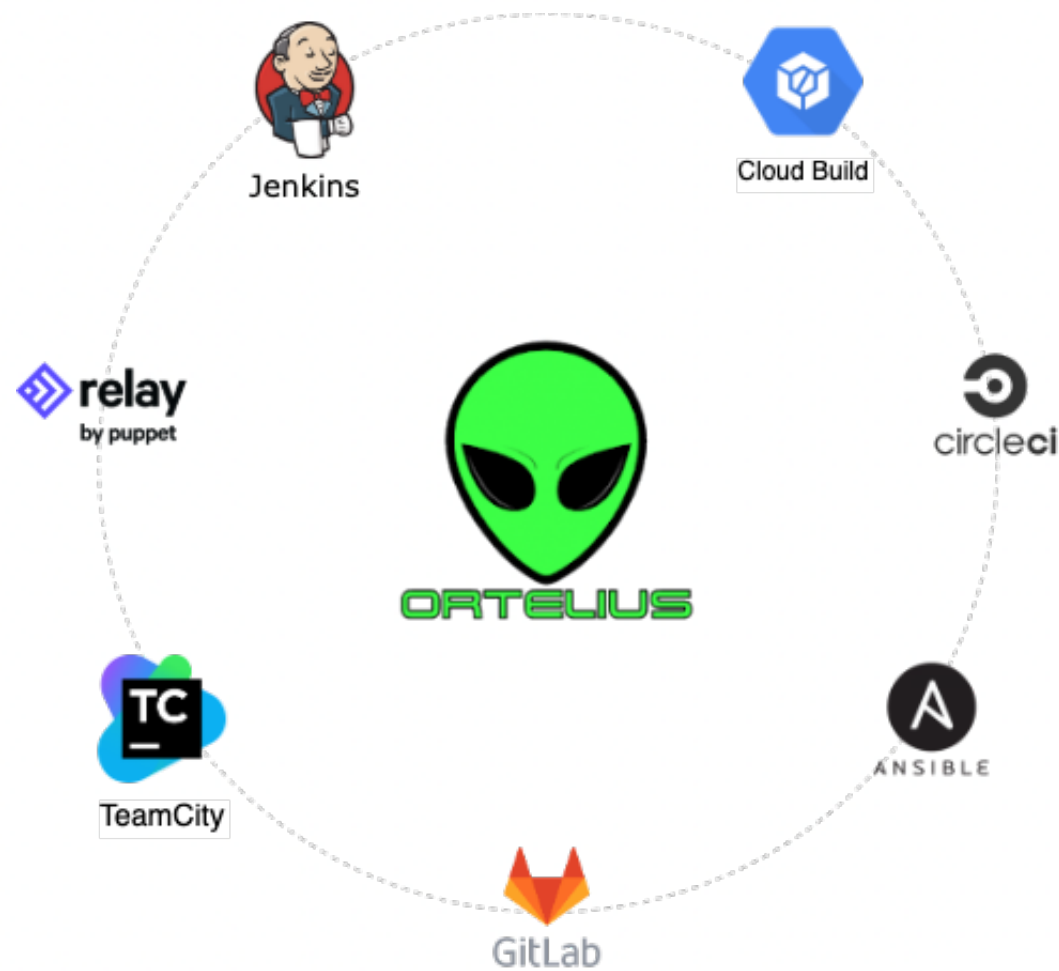
Domain Catalog



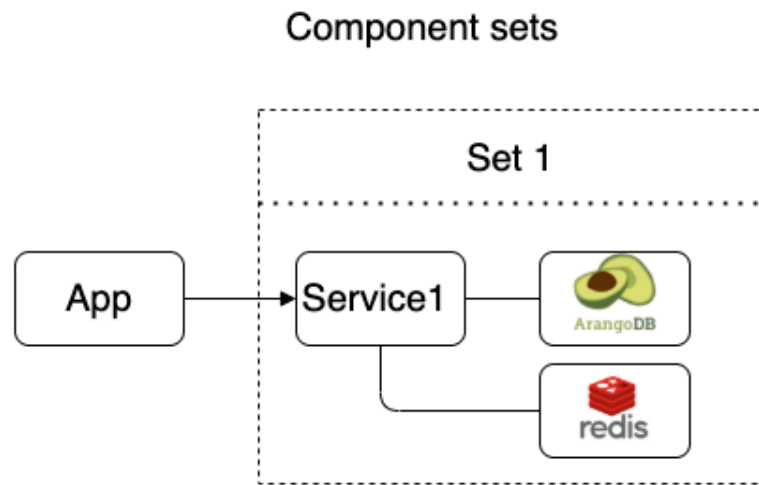
```

1 You, 5 months ago | 2 authors (You and others)
2 import routes from 'routes'
3 import httpService from 'services/core/httpService'
4
5 You, 5 months ago | 1 author (You)
6 class UserService {
7   public getUser(id) {
8     return httpService.get(`${routes.api.user}/${id}`);
9   }
10
11   public update = (property, config) => {
12     user[property].map(item => {
13       if (item.id === config.id) {
14         config.data.forEach(field => {
15           if (field.key.includes('[')) {
16             // ...

```

- Use CI/CD to automate catalog maintenance
 - Creation of new app/component versions
 - Records/executes deployments of new versions
- Ortelius provides:
 - Internal deployment capabilities
 - Integration with external CI/CD tools



Component usage summary

Component	Version	Application	Environment
Service 1	1.0	App1	Prod
	1.3	App1	Dev
	1.3	App2	Dev2

- Component sets
- Component usage summary
- Additional CI/CD integrations
- Dynamic Service discovery
- Integration with Open Telemetry
 - What changed that caused an error?

Key take aways





Questions ?



Thanks to the cdCon & all the conference sponsors



Appendix

-  Photo by [Food Photographer David Fedulov](#) on [Unsplash](#)
-  Photo by [Pranjall Kumar](#) on [Unsplash](#)
-  Image by [Gerd Altmann](#) from [Pixabay](#)
-  Photo by [Juan Encalada](#) on [Unsplash](#)
-  Photo by [Sebastian Herrmann](#) on [Unsplash](#)