

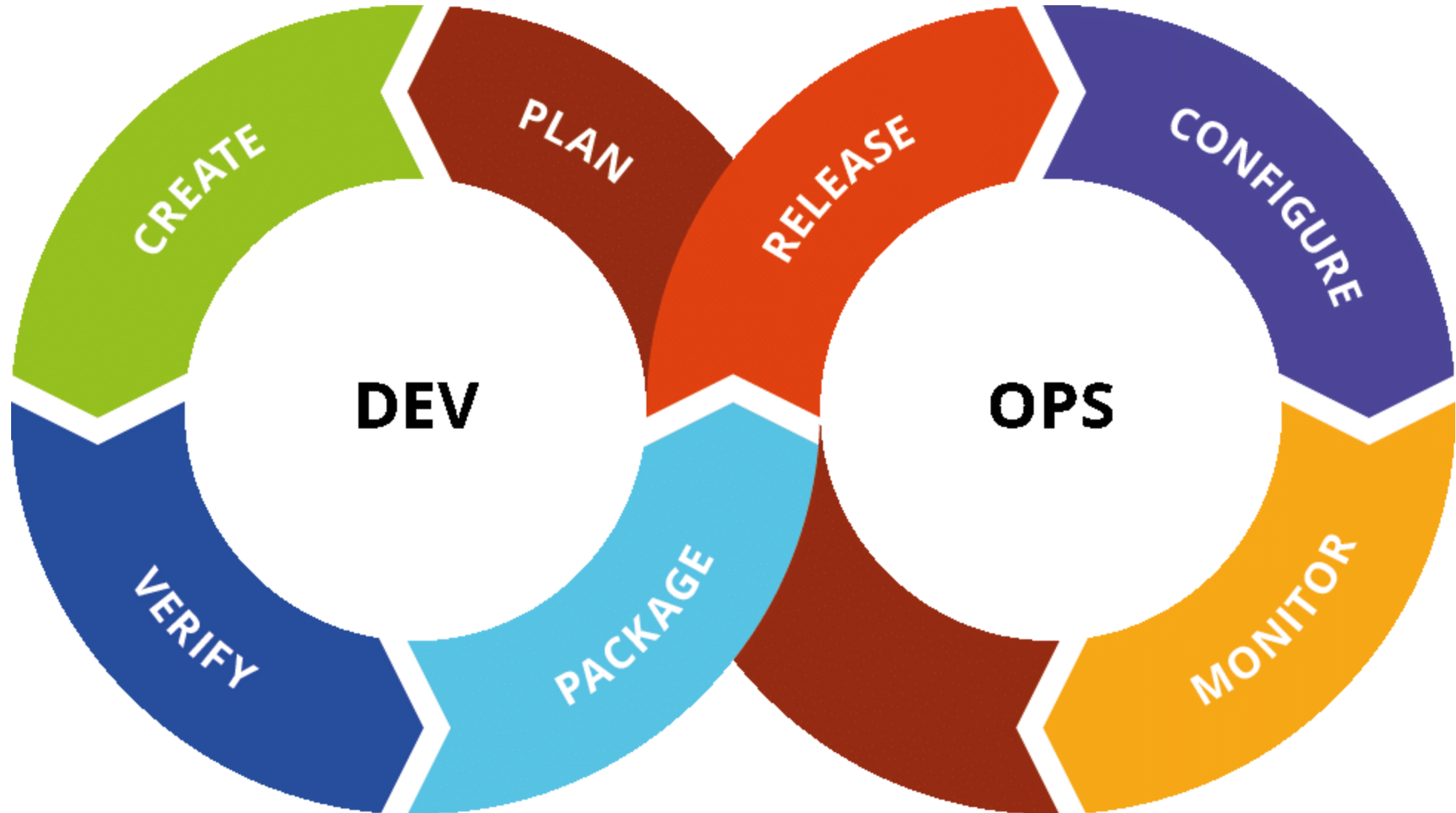
Azure Cloud Application Development Modern DevOps

Valdemar Zavadsky

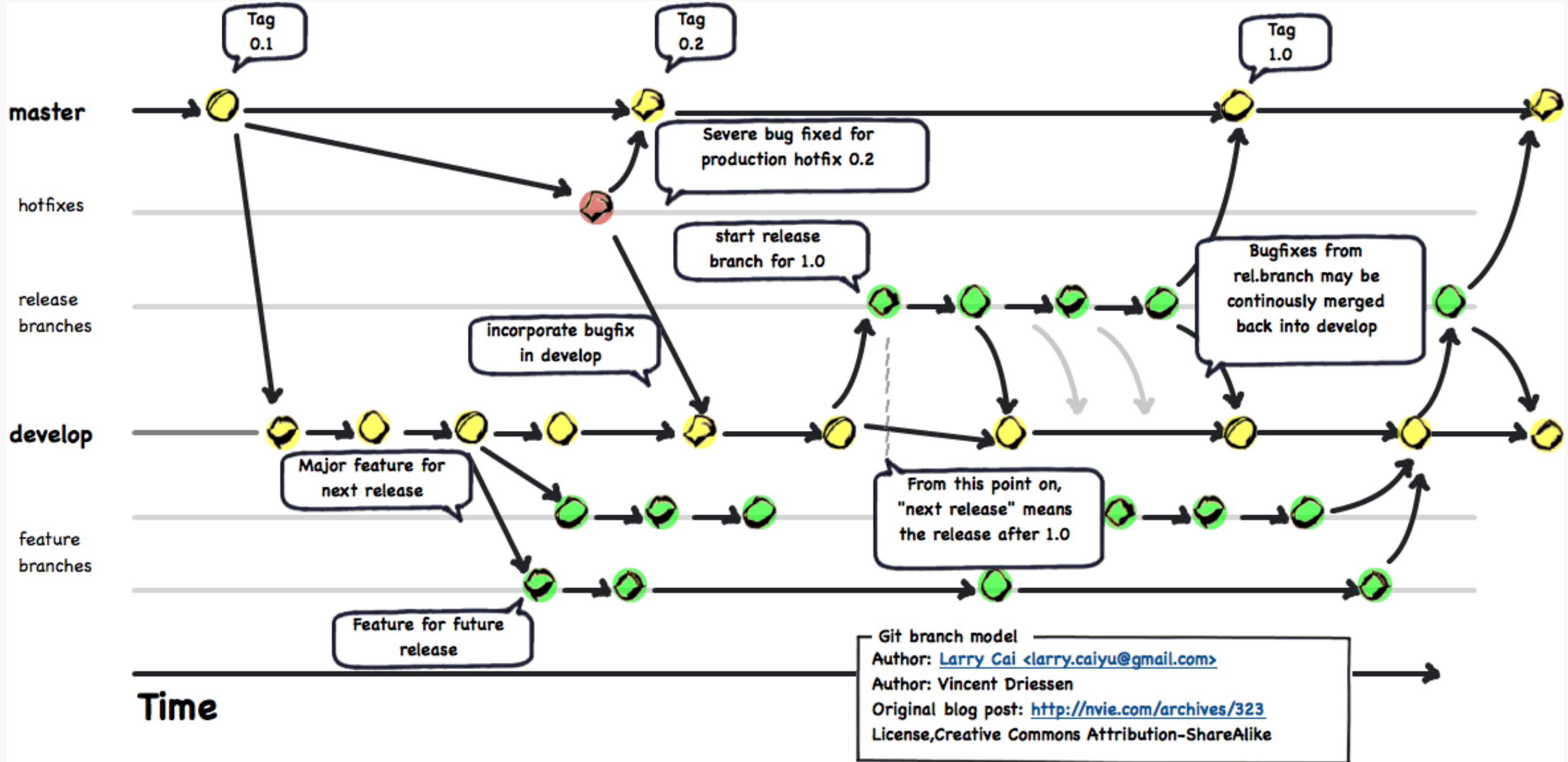
Cloud Solution Architect

Microsoft

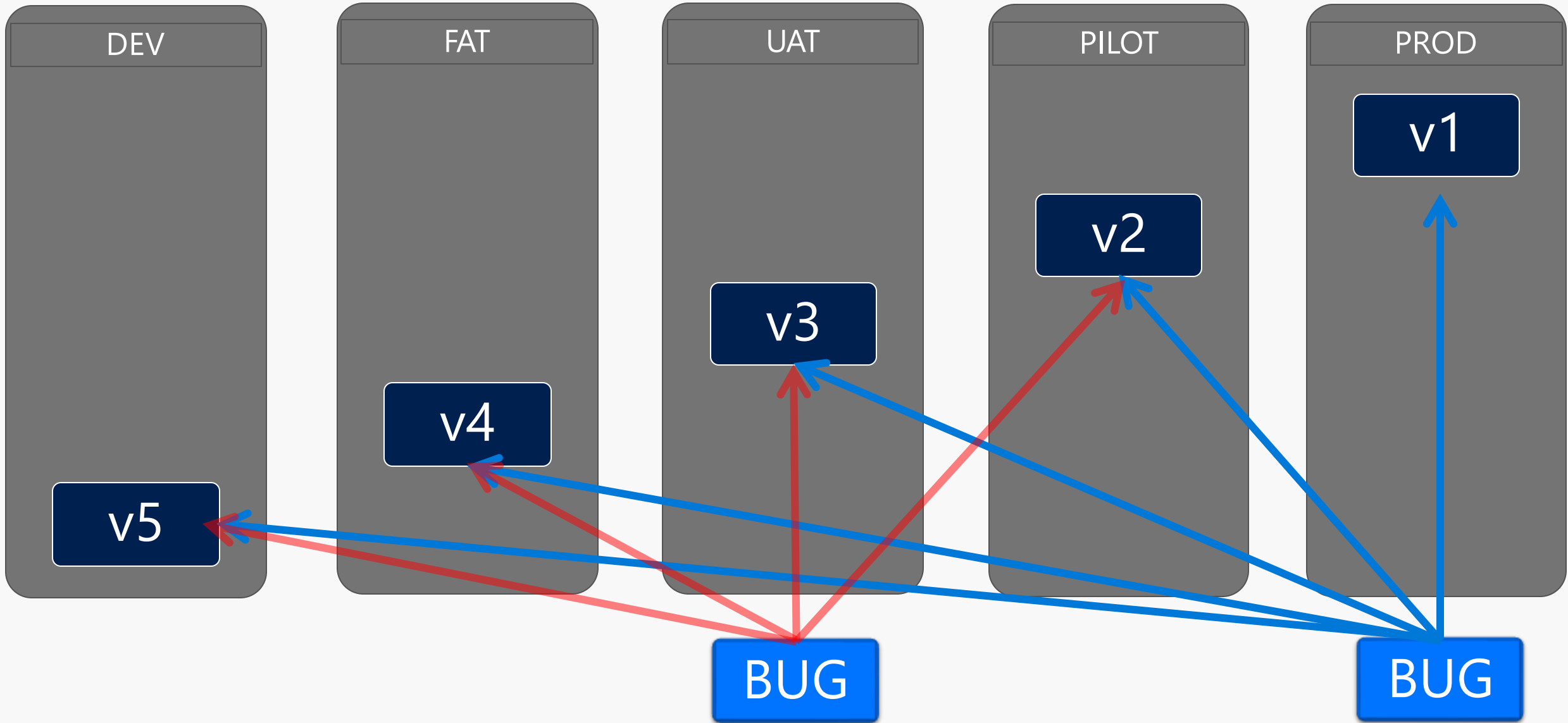
DevOps



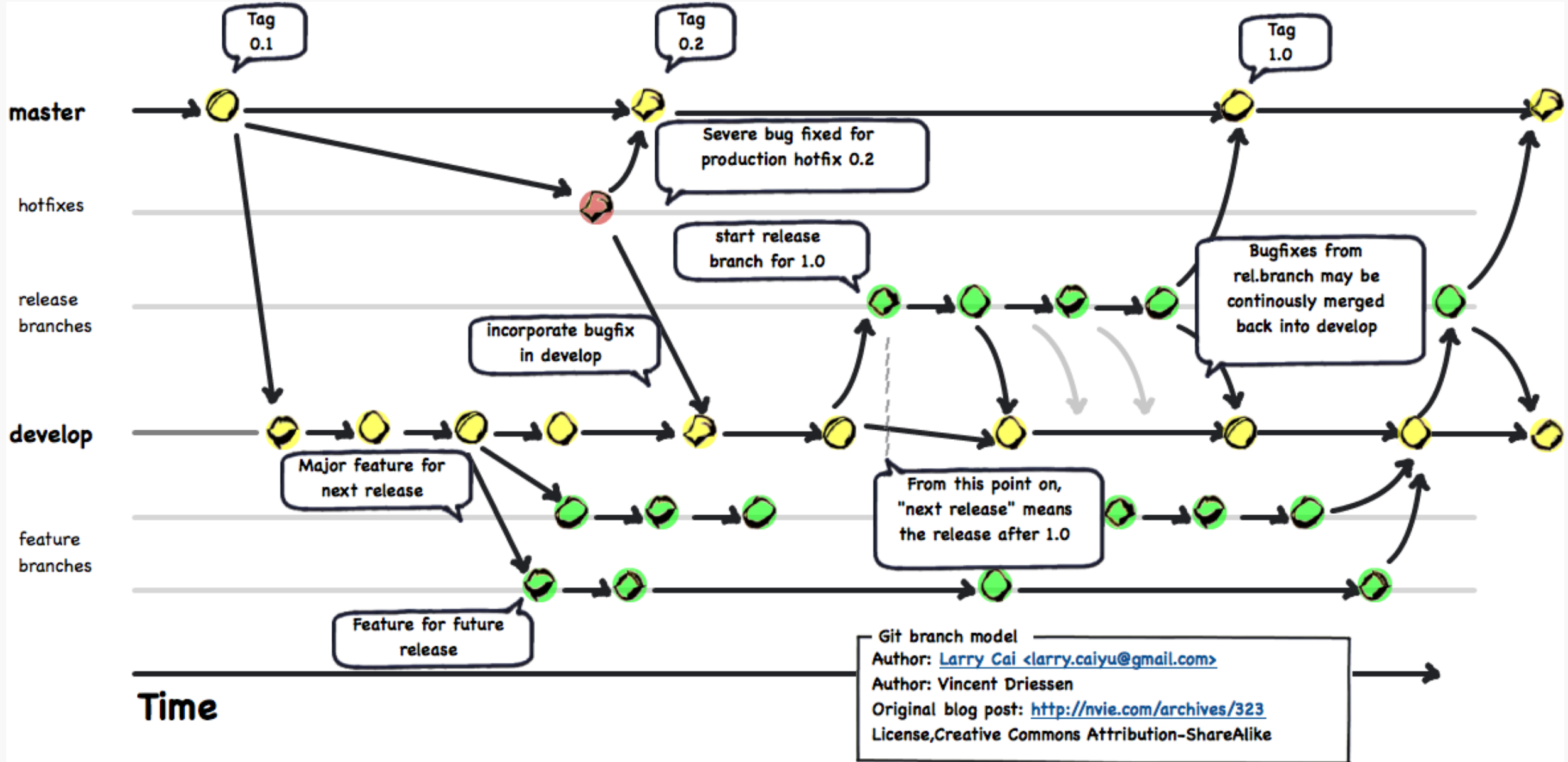
Source code organization



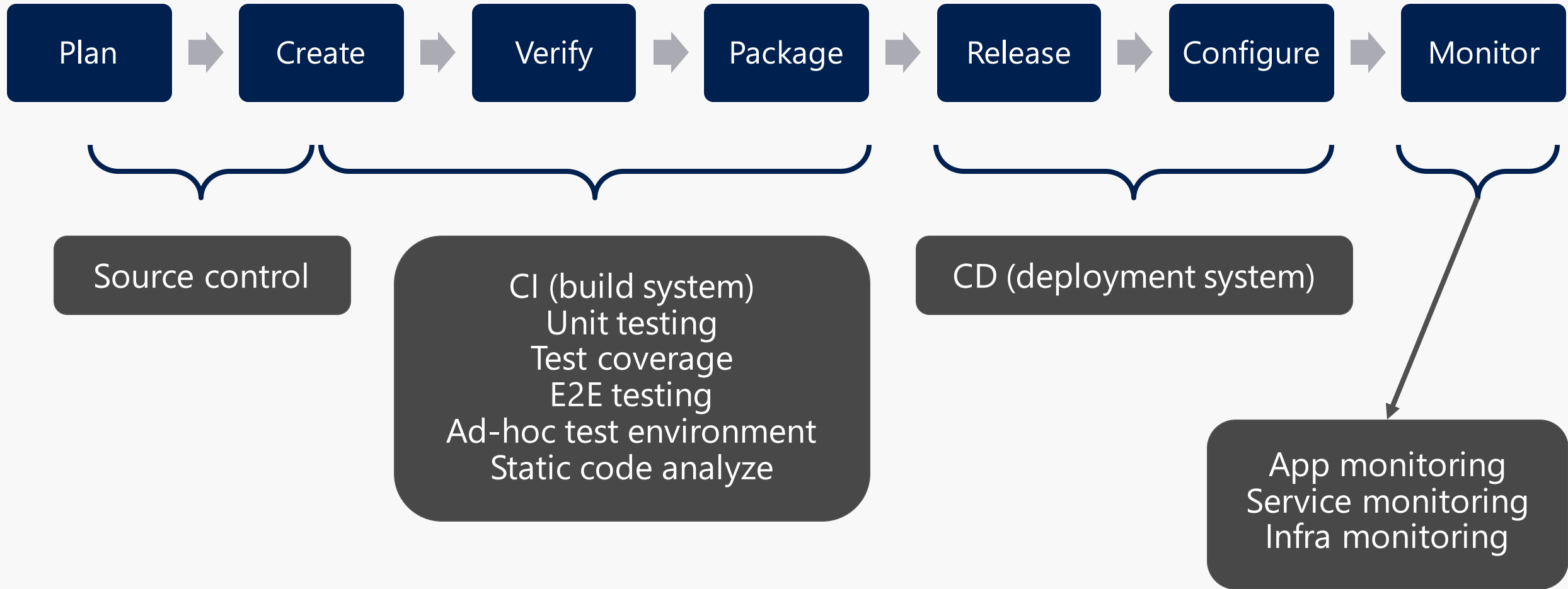
Why?



Source code organization



DevOps



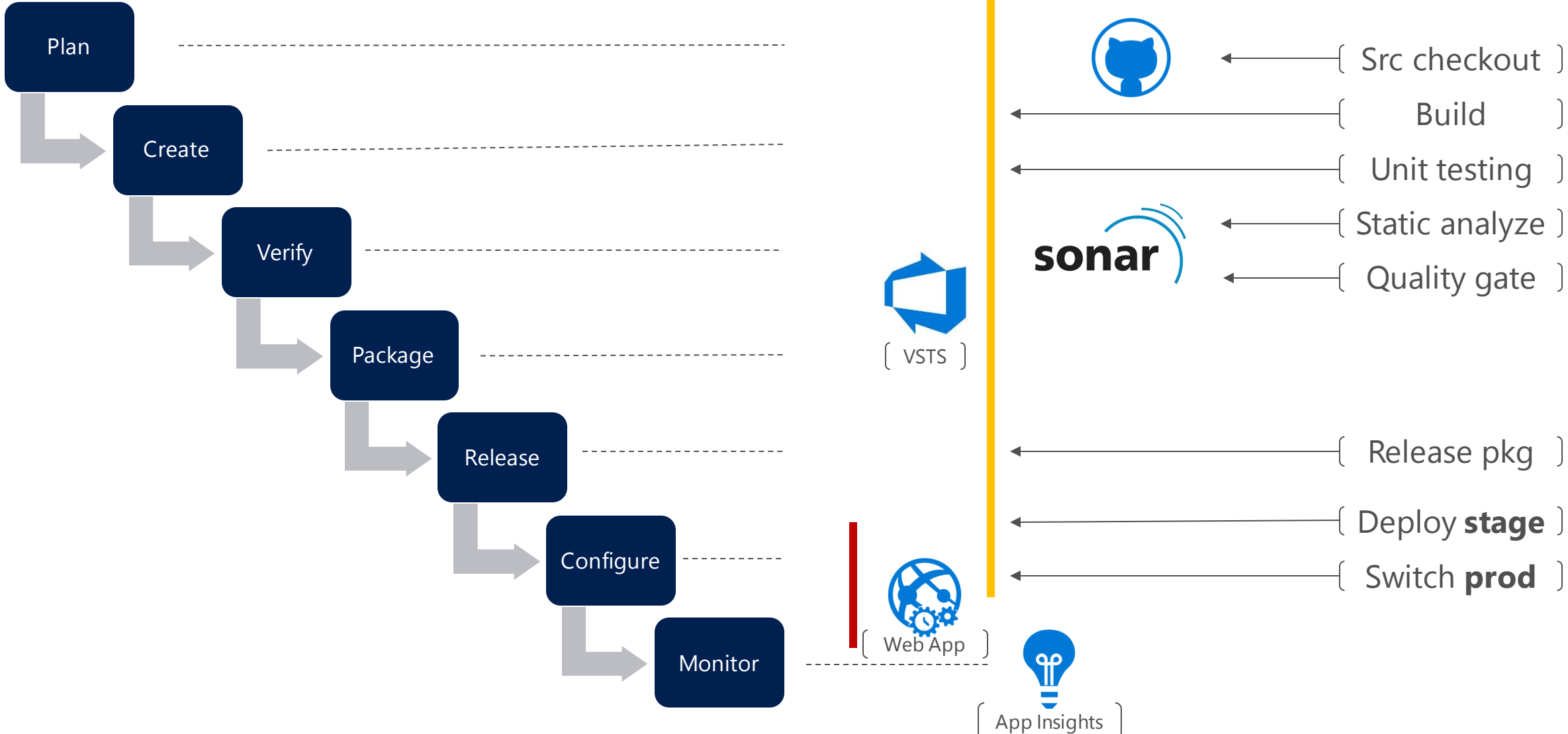
Dark side of DevOps



- DevOps is not simply combining Development & Operations teams
- DevOps is not a separate team
- DevOps is not a tool
- DevOps is not a one-size-fits-all strategy
- DevOps is not automation

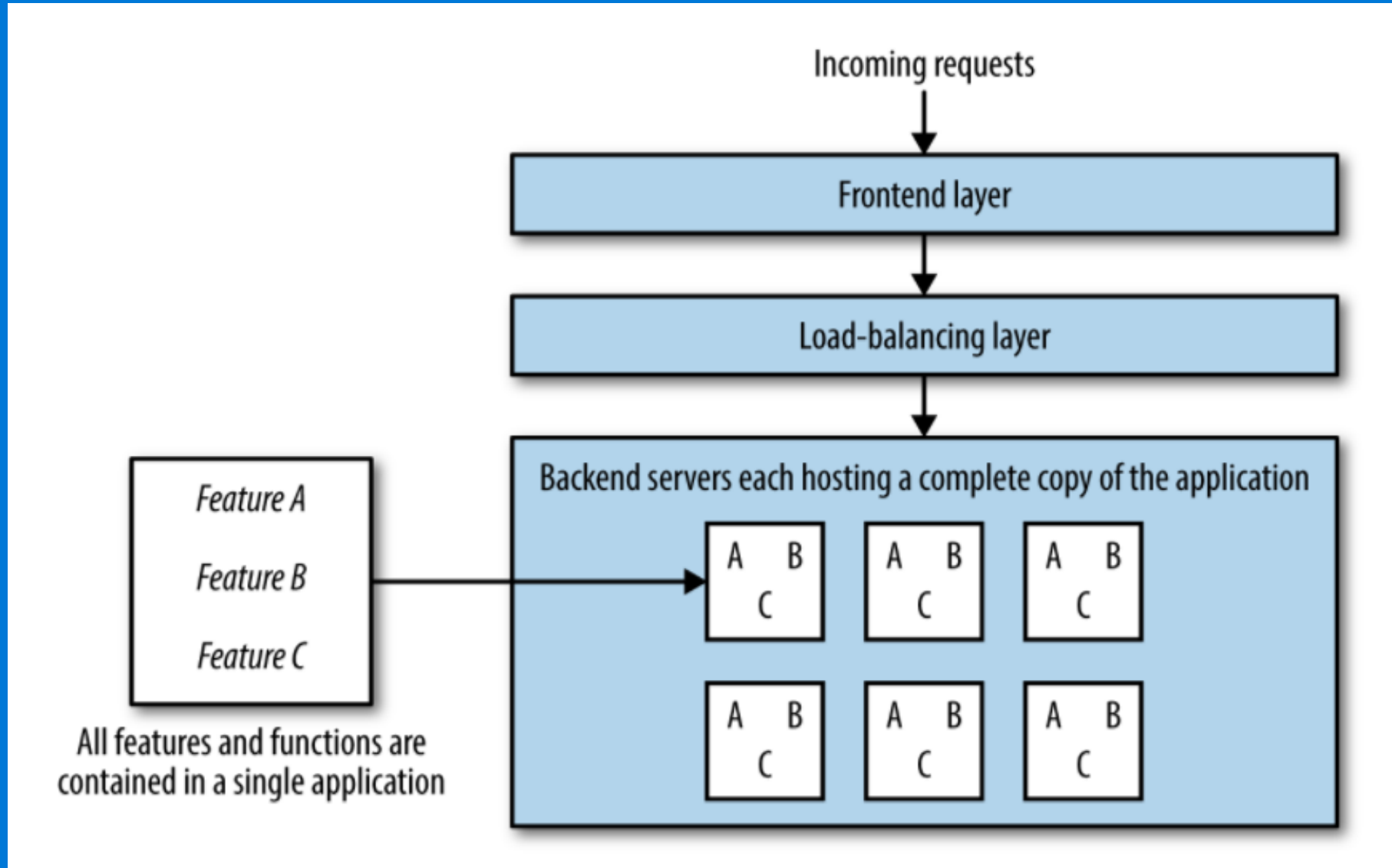
C# / VSTS / Azure PaaS (WebApp, SQL)

CI/CD pipeline

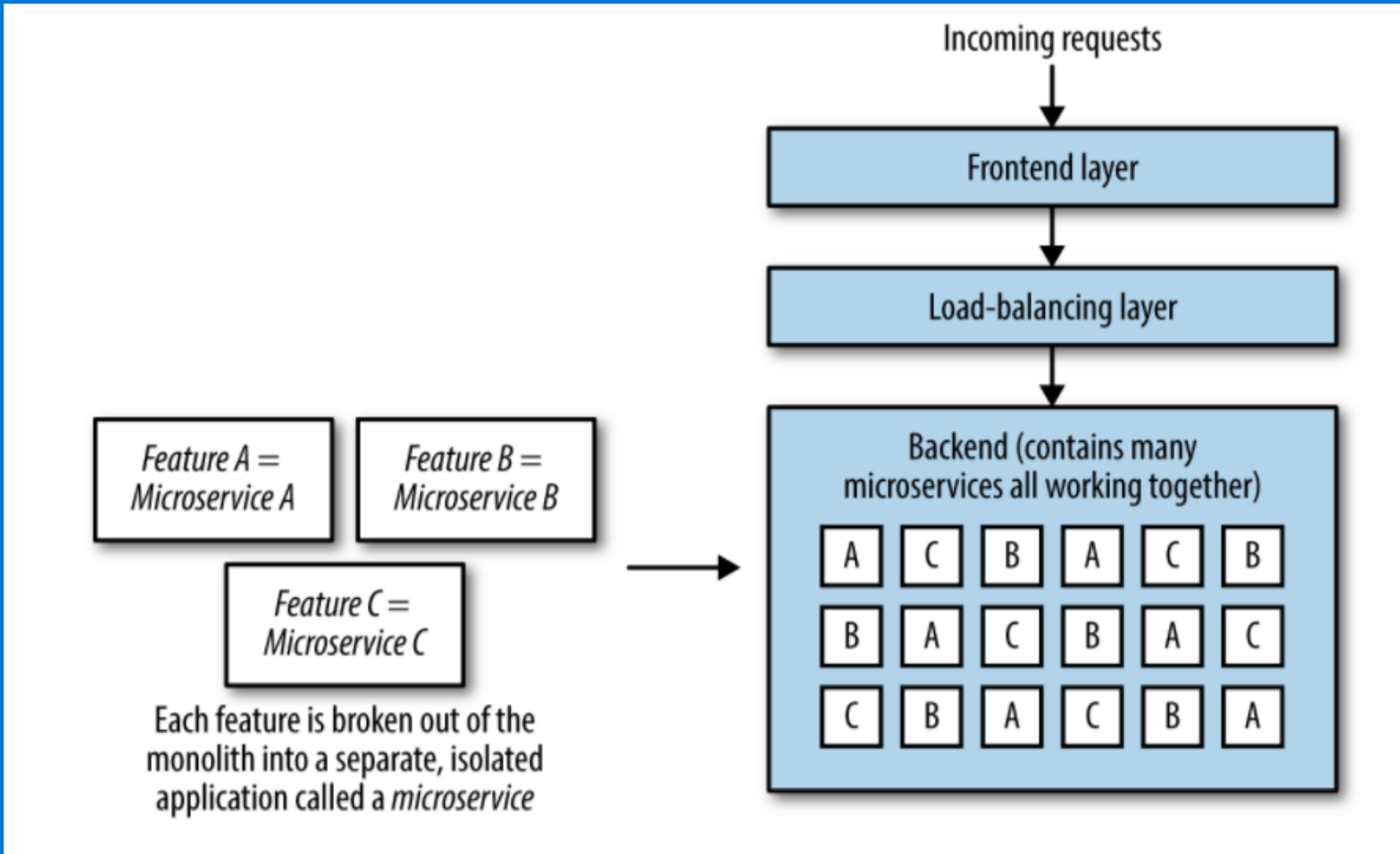


Microservices in 5 minutes

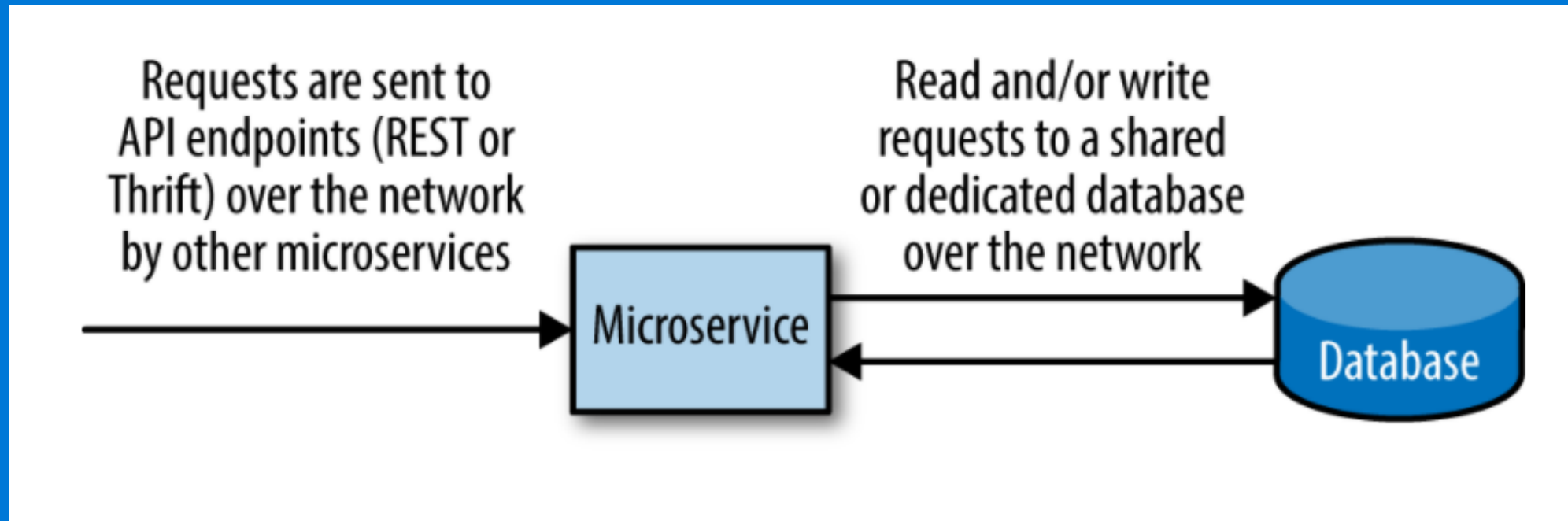
Monolithic app



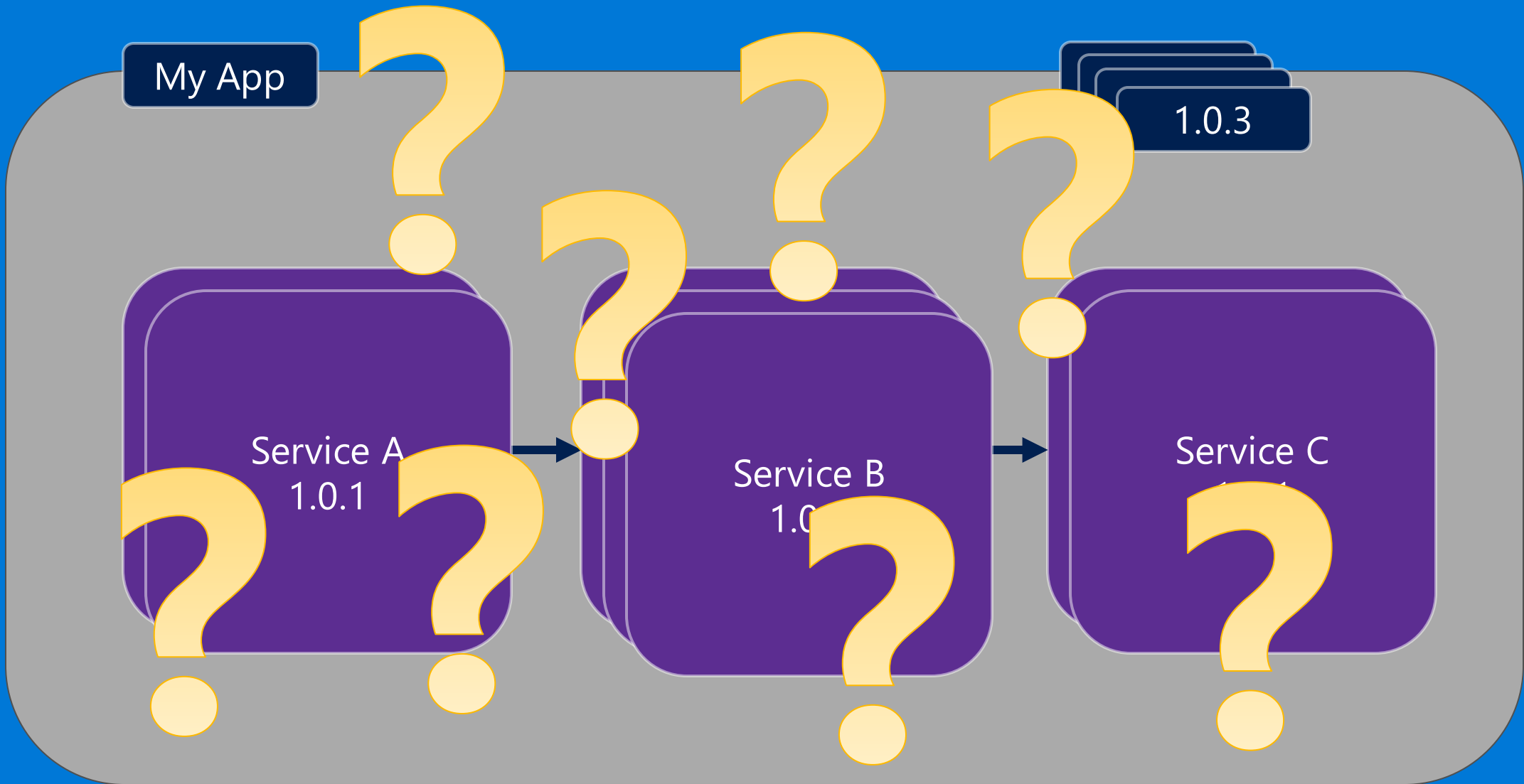
Microservices ~~app~~



Microservice



Microservice ~~versioning~~



Microservice ~~versioning~~

AVOID VERSIONING MICROSERVICES AND ENDPOINTS

A microservice is not a library (it is not loaded into memory at compilation-time or during runtime) but an independent software application. Due to the fast-paced nature of microservice development, versioning microservices can easily become an organizational nightmare, with developers on client services pinning specific (outdated, unmaintained) versions of a microservice in their own code.

Microservices should be treated as living, changing things, not static releases or libraries. Versioning of API endpoints is another anti-pattern that should be avoided for the same reasons.

Microservices contract versioning

<https://github.com/Microsoft/api-guidelines/blob/master/Guidelines.md#12-versioning>

BREAKING . FEATURE . FIX

**incompatible
API changes**

**breaking
change**

**add backwards-
compatible
functionality**

**new
feature**

**make backwards-
compatible bug fix**

**bug
fix**