



RELEASES, VERSIONS, BRANCHES.

A Modest Proposal For preventing the Software of Average Developers From being a Burden to Their Manager or Organization, and For making it Beneficial to the Public.

RELEASES

The association of a validation level with a software artifact

A released artifact is immutable

And is identified through a version number



VERSIONS

semantics incremented when ...



x.y.**z** patch backwards compatible bug fixes are introduced

x.**y**.z minor new, backwards compatible functionality is introduced

x.y.z major backwards incompatible changes are introduced

WHERE IS THE VERSION INFORMATION ?

external to the artifact

+ embedded within the artifact

+ in the source code as well



BRANCHES

"Branching is the practice of creating copies of programs or objects in development to work in parallel versions" 

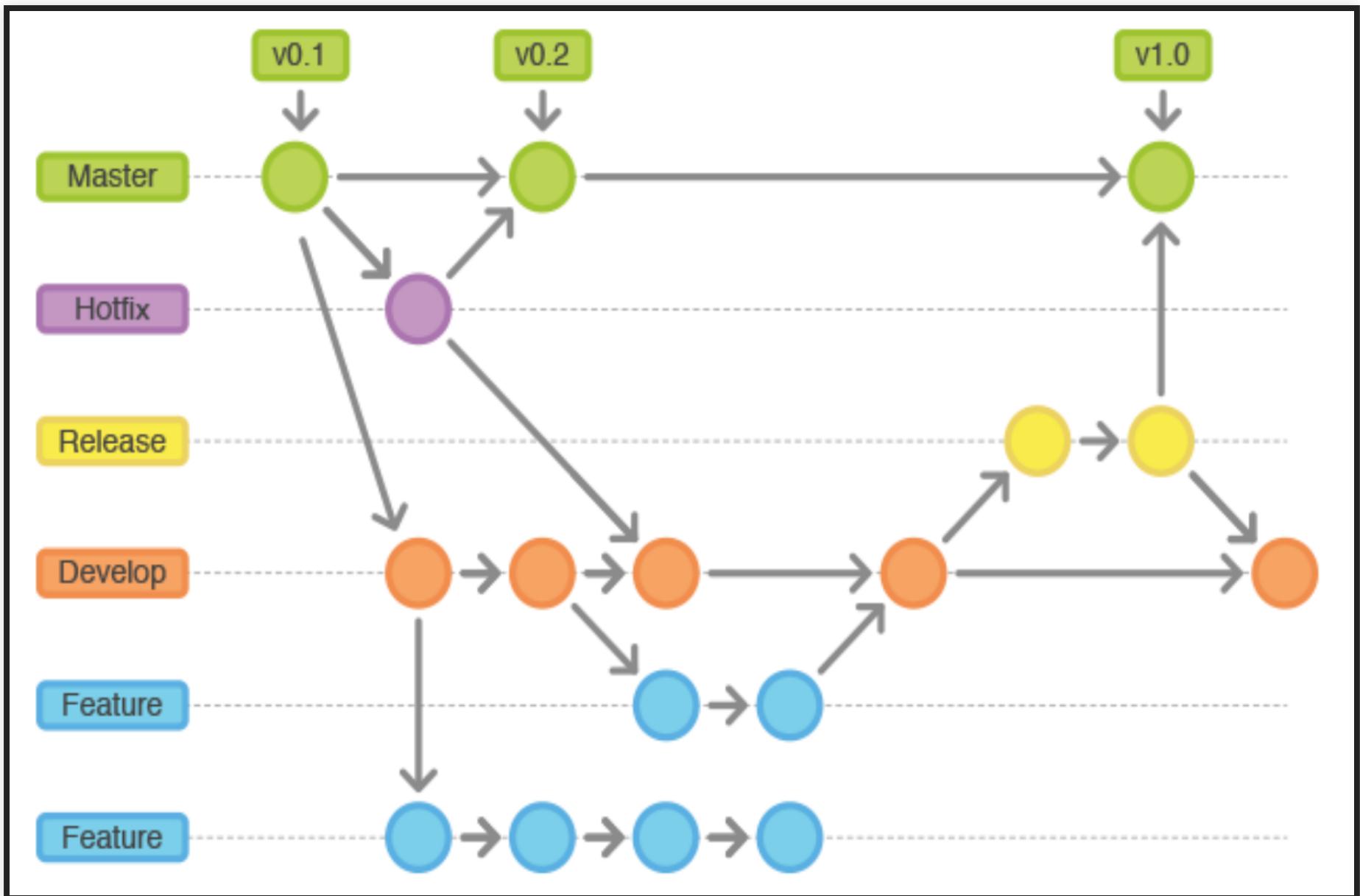
Creating a branch is cheap, merging it back to trunk is expensive

DEMO TIME

GITFLOW

"The Gitflow Workflow defines a strict branching model designed around the project release" 

Original post by [Vincent Driessen](#)





ON-PREMISE VS. CLOUD

Do we need all these branches for an application
operated in the cloud?

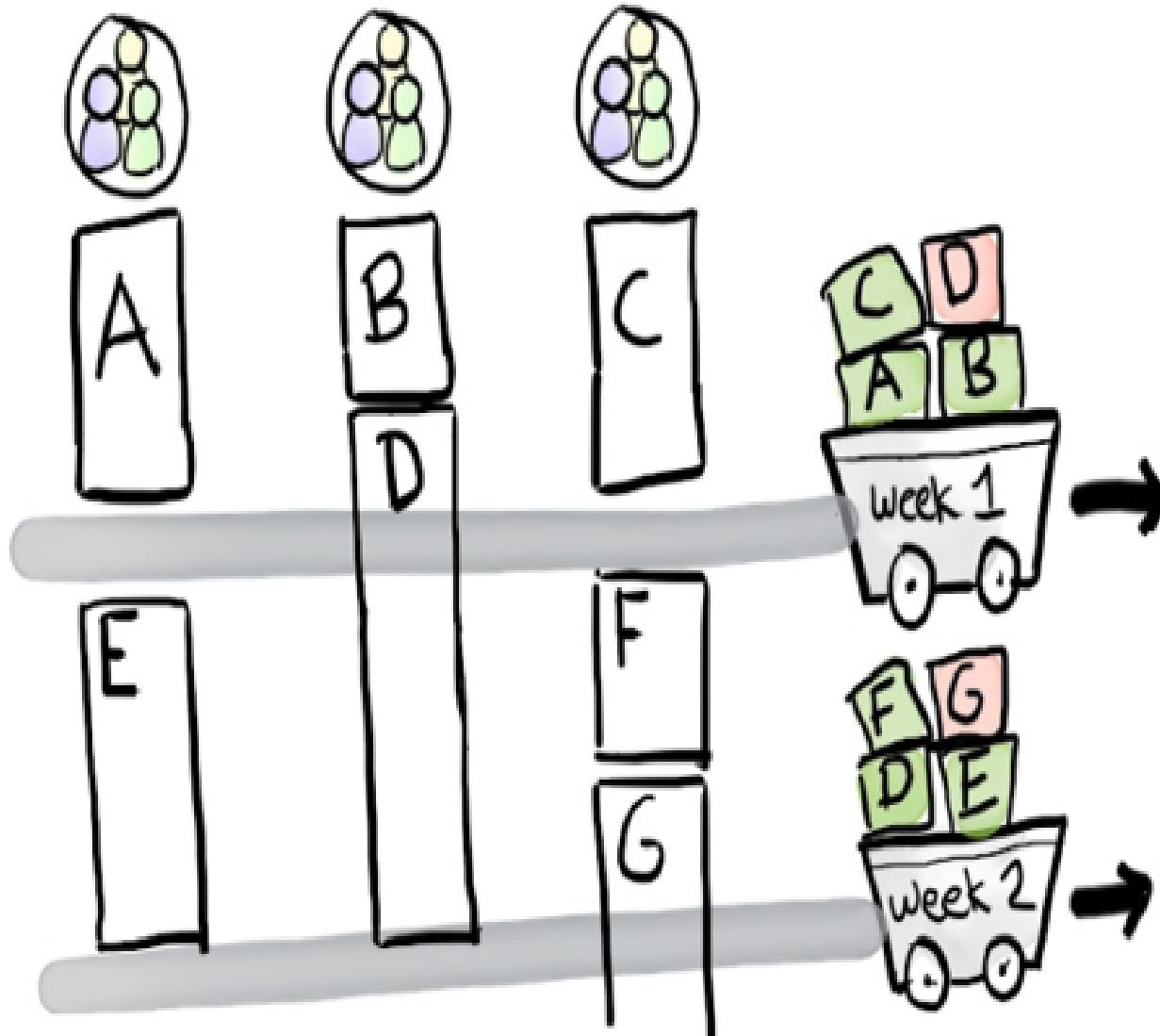
One URL → One App of One Version → One Branch!

 **FEATURE TOGGLE**

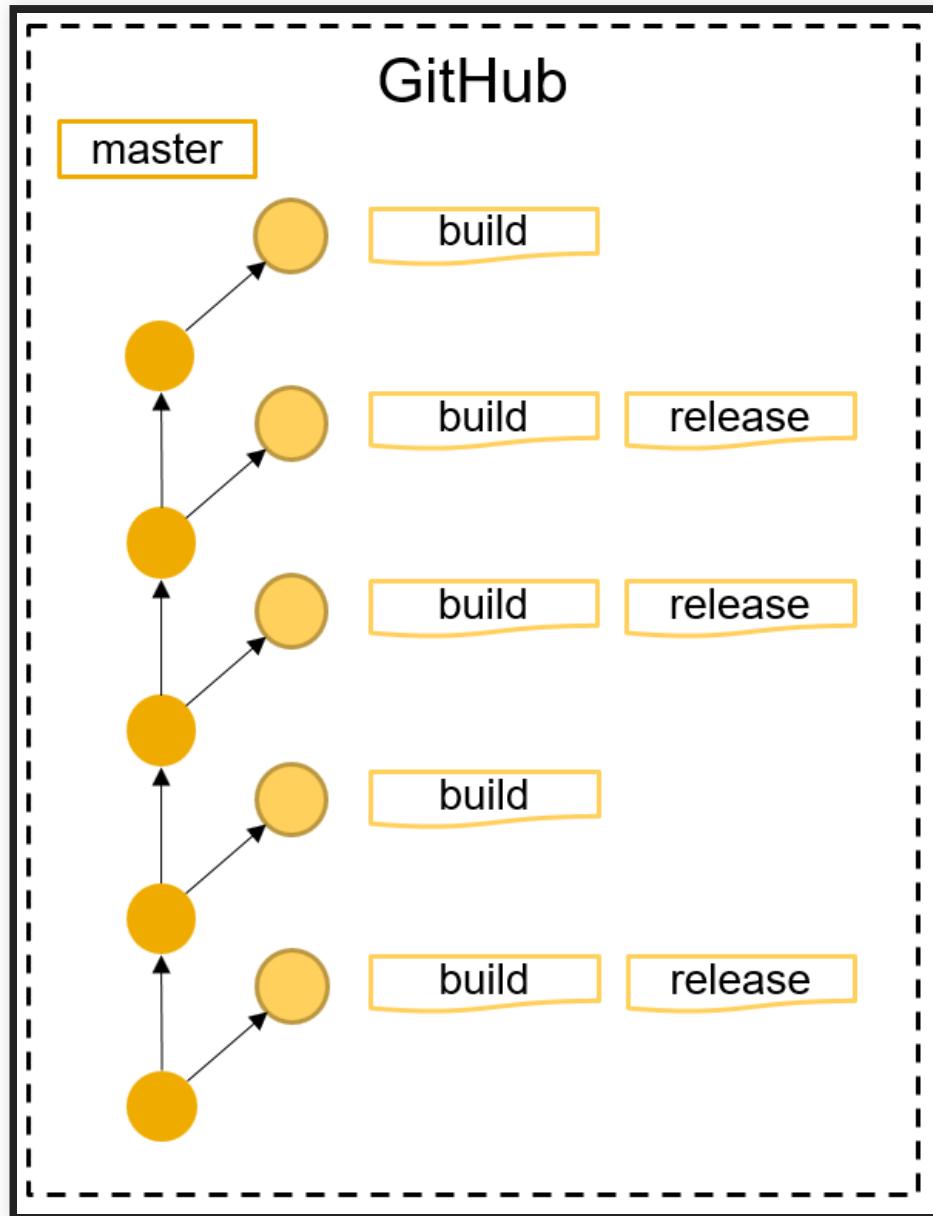
"a technique in software development that attempts
to provide an alternative to maintaining multiple
source-code branches" 

Visibility

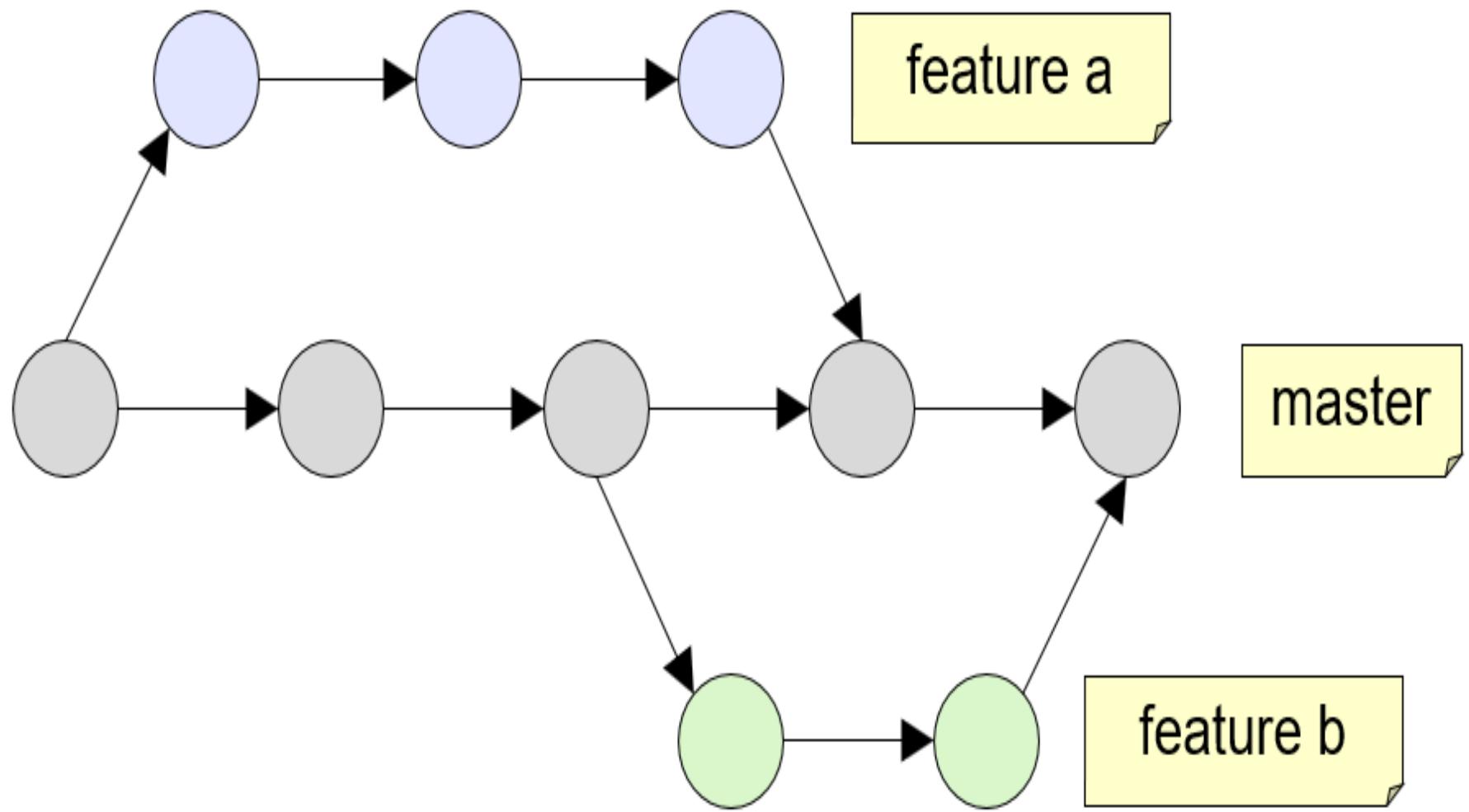
- Feature A
- Feature B
- Feature C
- Feature D

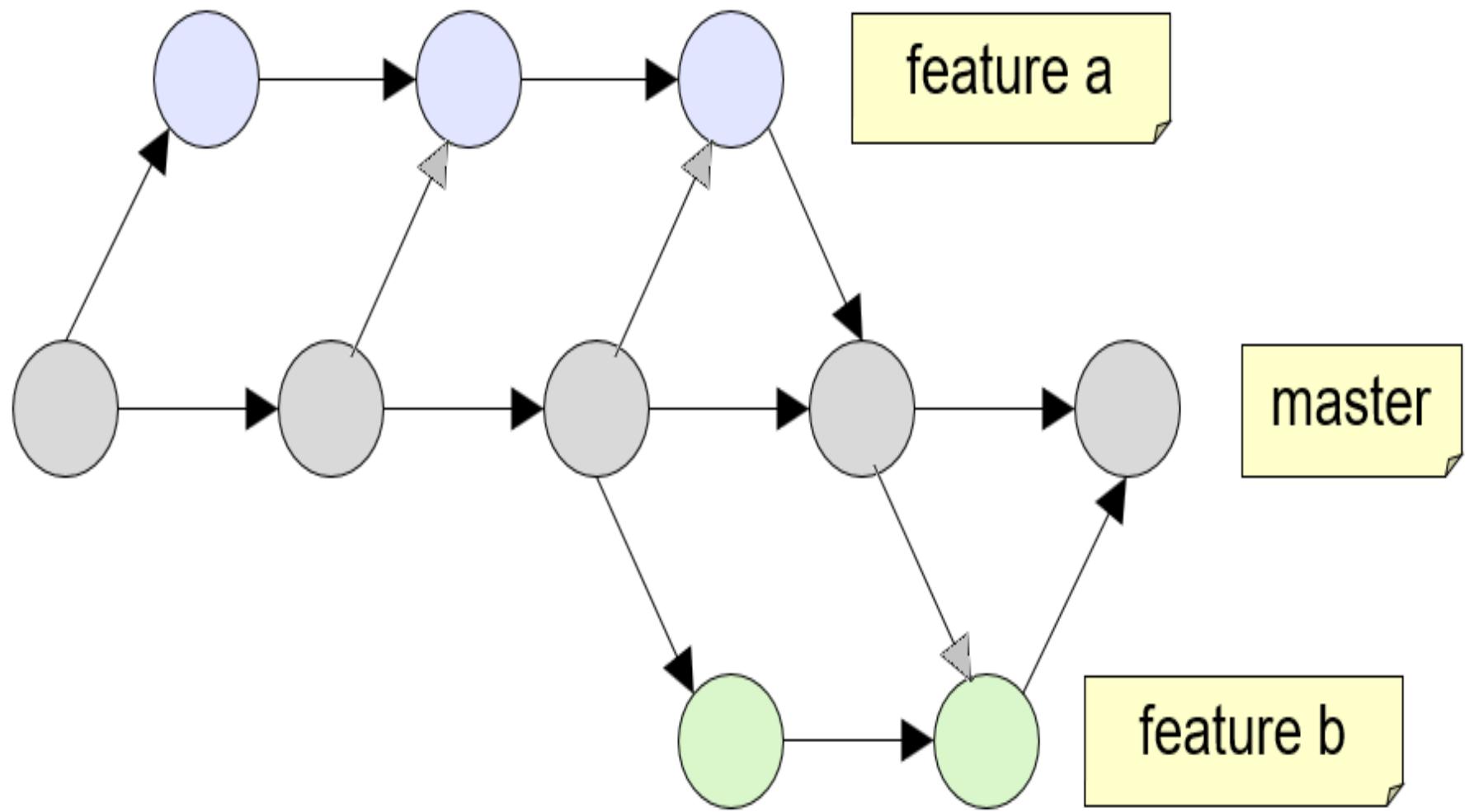


Apply the 'One Branch' mantra to the Release Process



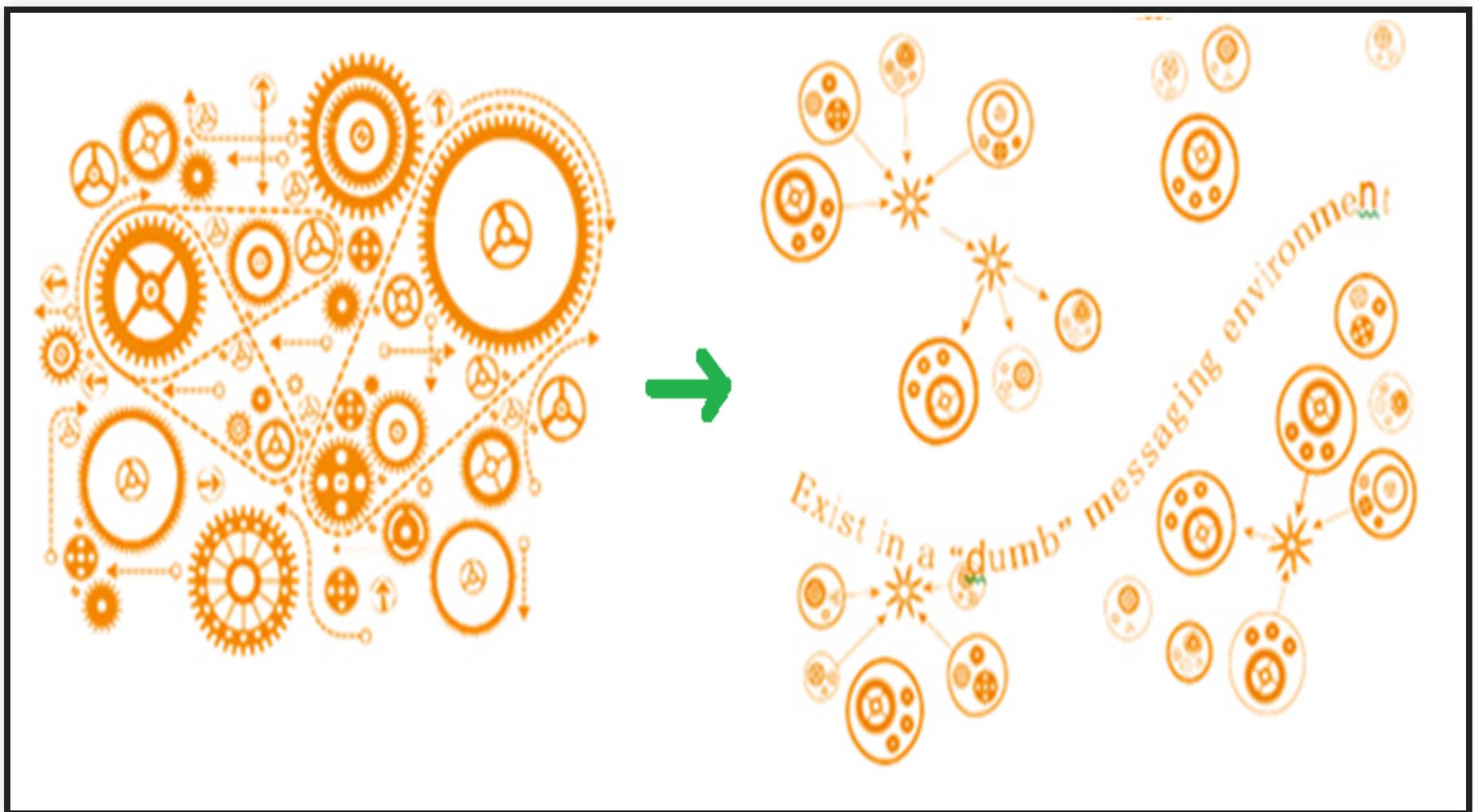
Are Feature Branches that bad?





Decoupling

from Monolith (tightly coupled) to Microservices (decoupled)



Two microservices and their shared database.



QUESTIONS?

Learn French.
It is much
easier than
to understand
French speaking
English.

BELECOLE

Apply your French lessons to the most French place in Prague.
More at www.belecole.cz