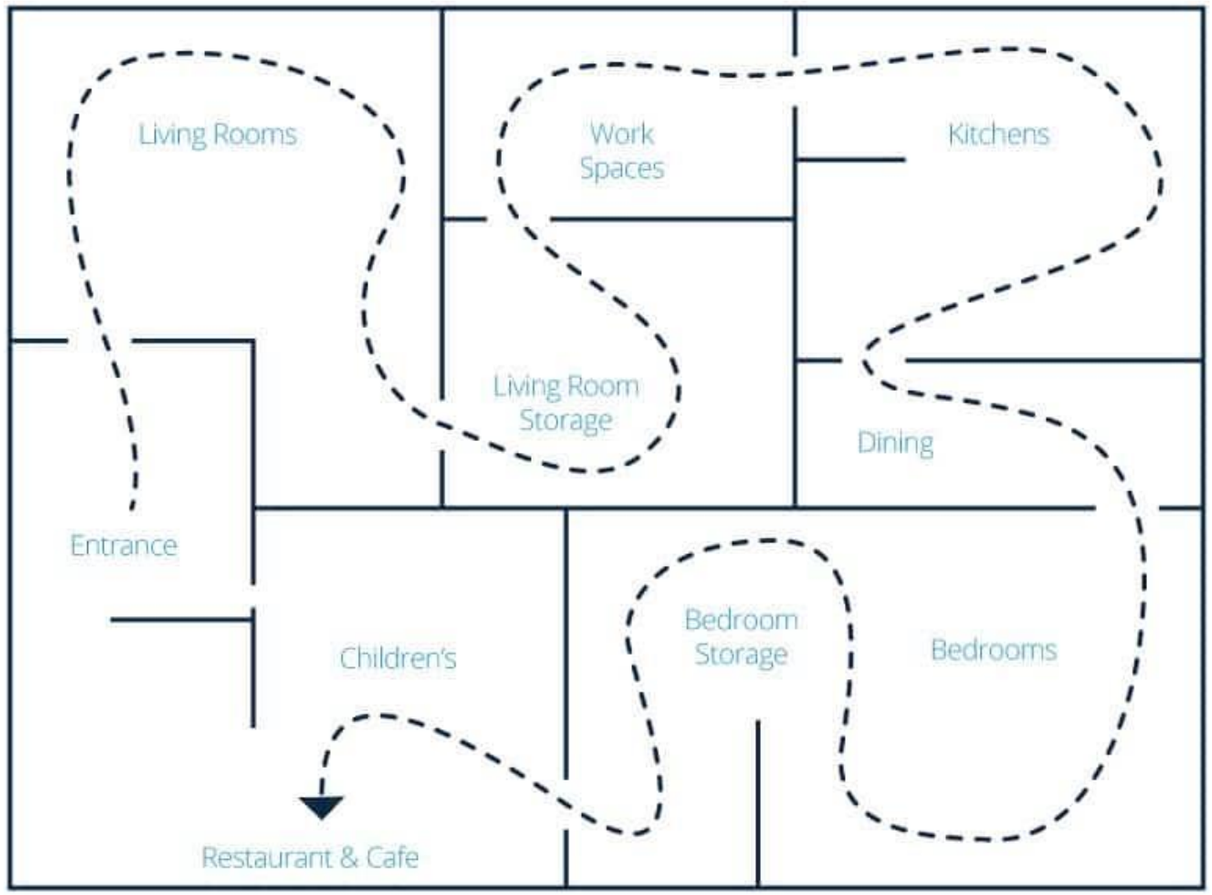
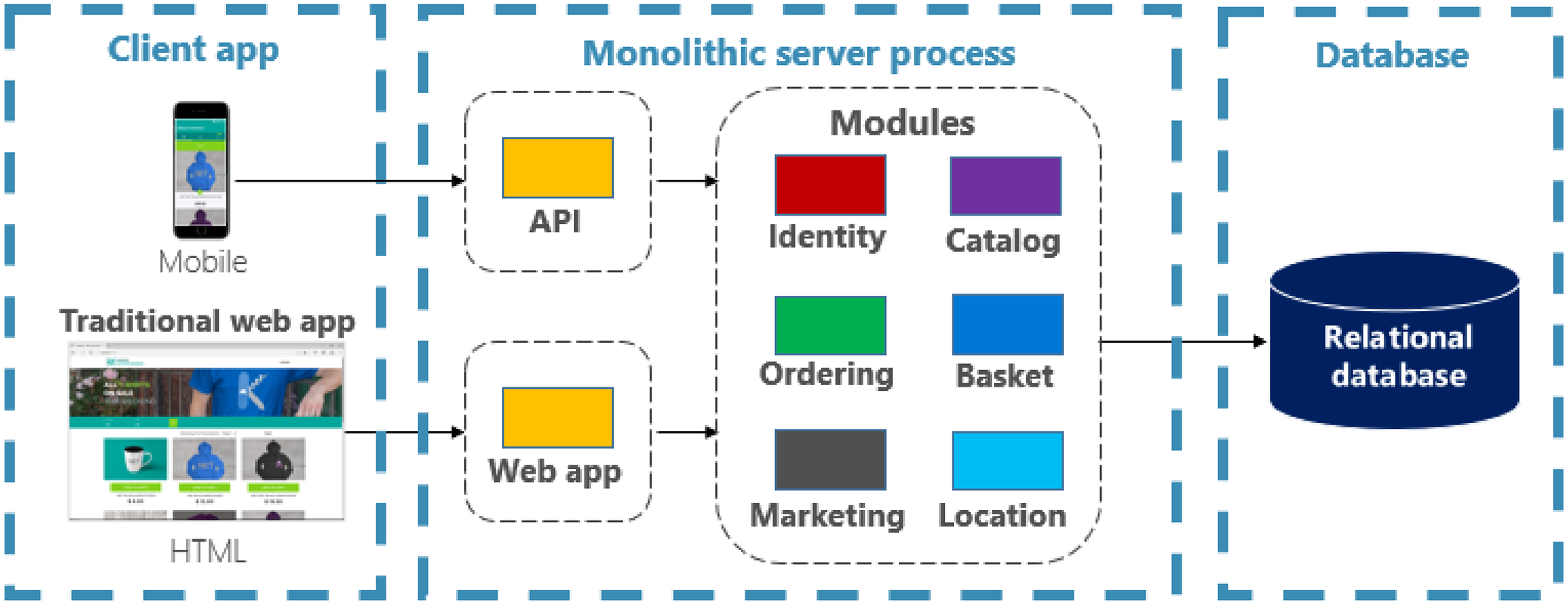
Developer Operations (devops)

Application Architecture and Conventions for DevOps

Convention over Configuration

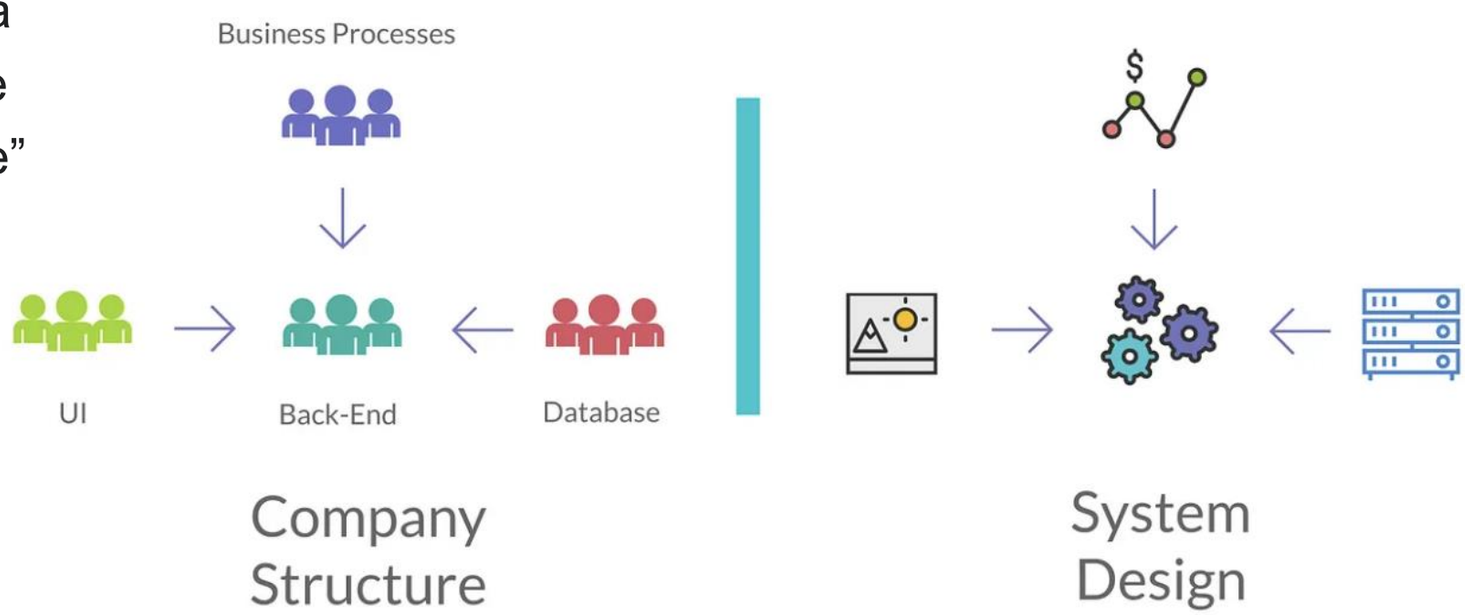


What are the problems with this architecture?

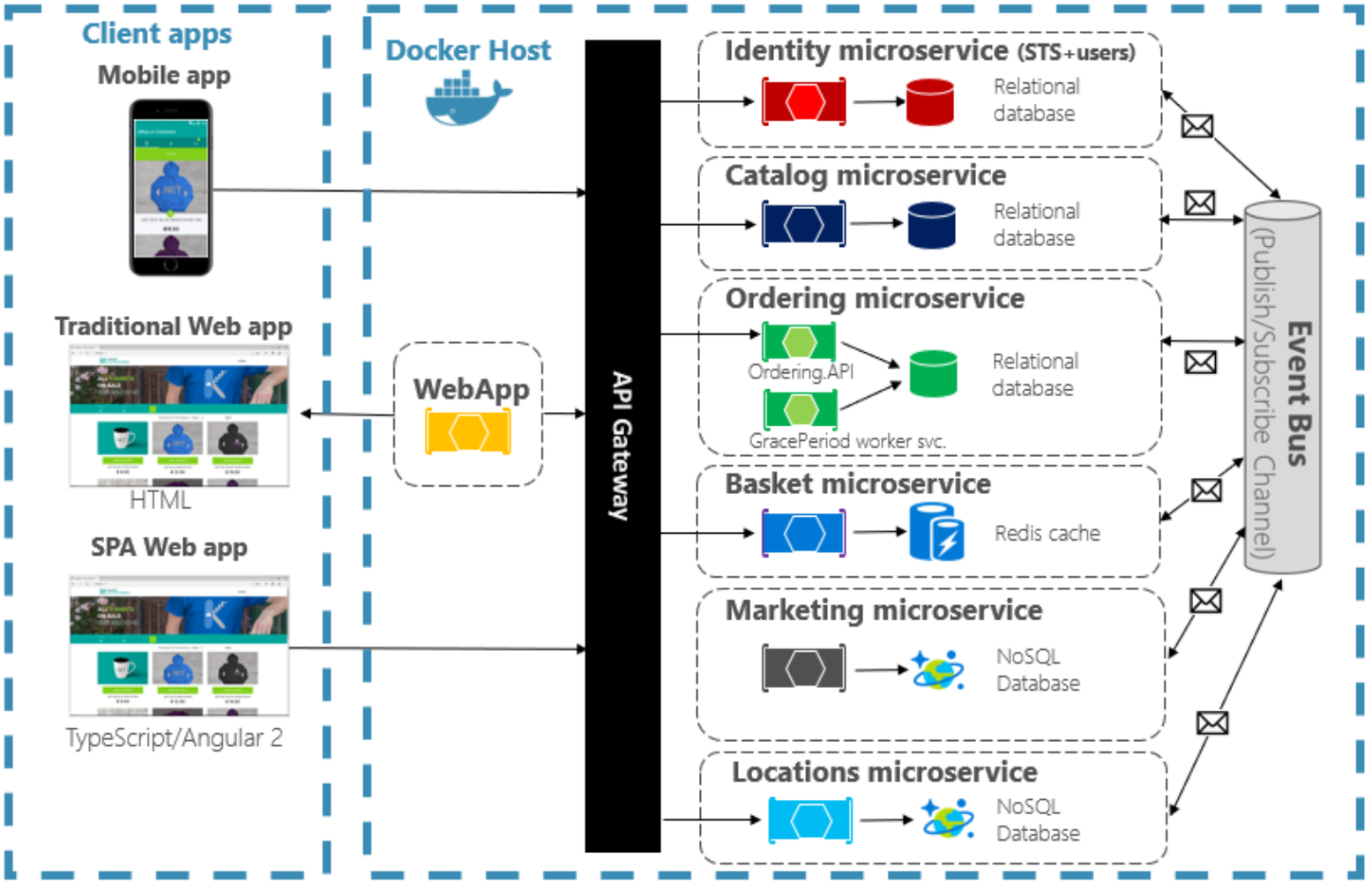


Conways Law

- “Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure”
- - Melvin E. Conway



Smaller Services, aligned with value stream



12factor-App

1. One Codebase
2. Active Dependency Management
3. Environment configurations separate from application
4. Use backing services as external resources
5. Handle build, release and run as separate tasks
6. Execute the app as stateless processes
7. Communicate via fixed ports only
8. Use process model for scalability
9. Start fast and shutdown graceful
10. Keep stages similar
11. Logs are event streams
12. Admin/Mgmt are one-use processes

https://12factor.net/

One Codebase, Keep everything in Sourcecode

All data necessary to build and deploy you app should be in the version control:

- Sourcecode
- Functional Configuration e.g. Database Connection
- Icons
- Stylesheets
- Non-Functional Configuration e.g. Logging

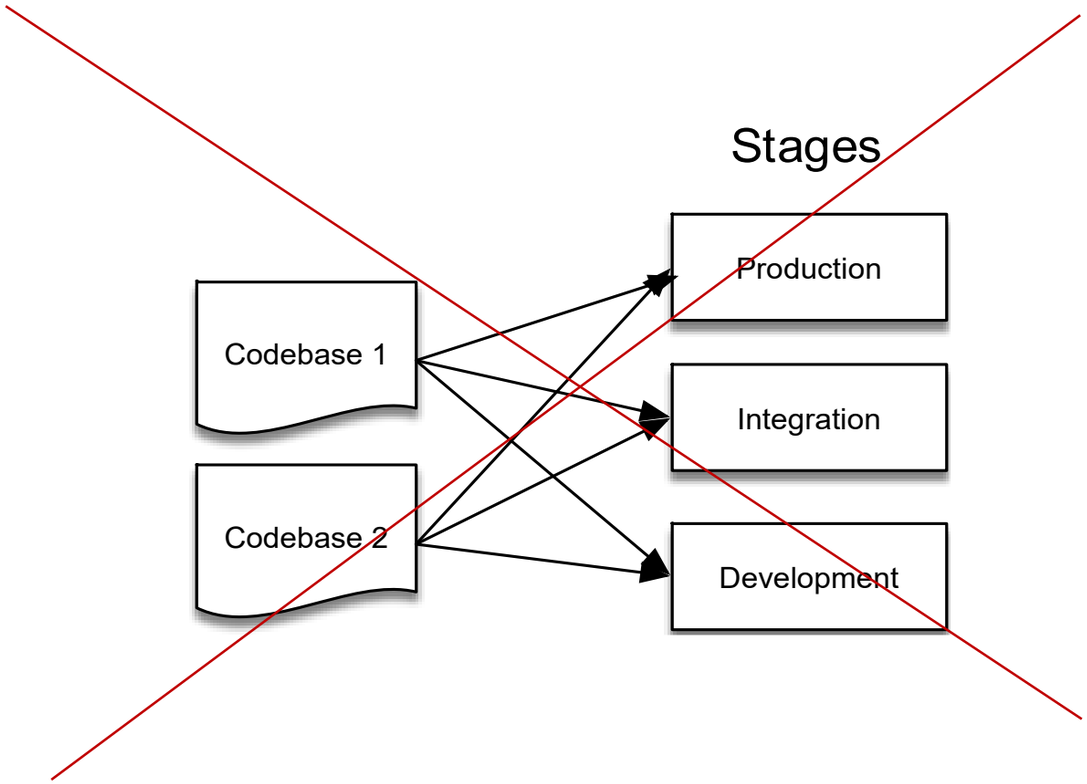
If you have large binary files necessary for build and if you can't use externalized storage (due to atomicity, etc.) → Git LFS

Everything generated should NOT be in the source code:

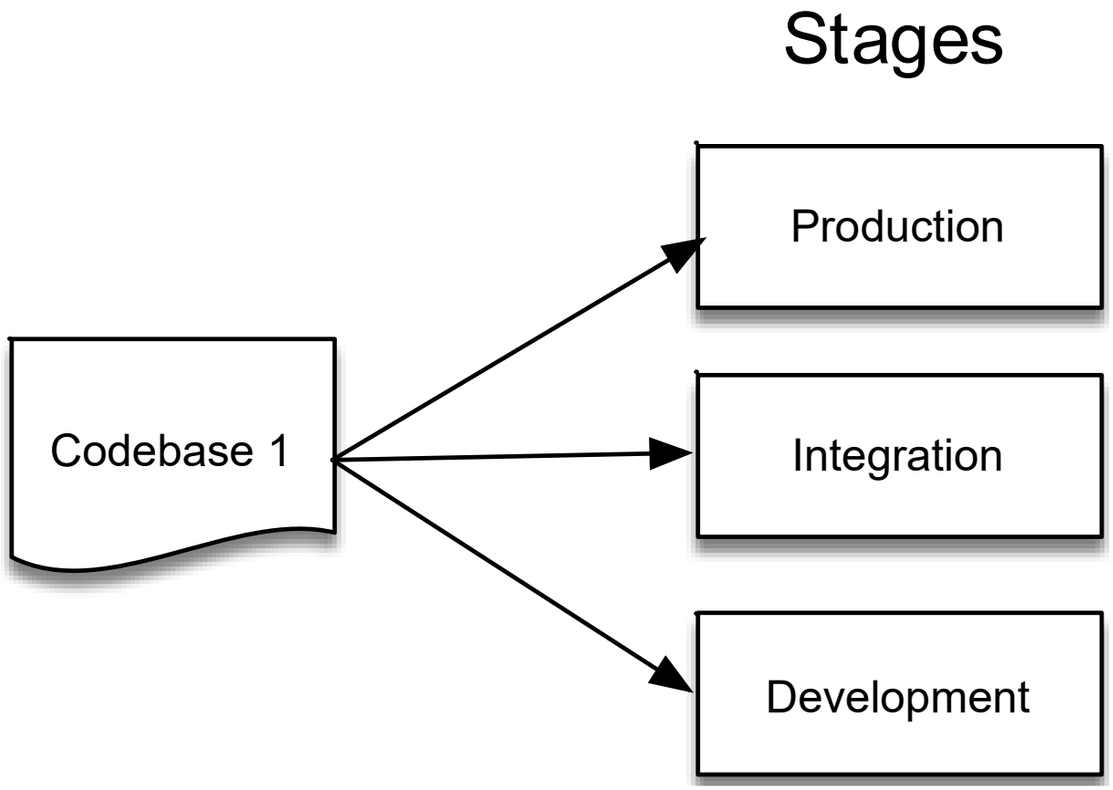
- Generated jar/war/binaries
- Generated Docs / Websites
- node_modules / target
- Generated testdata

Use an artifact repository (Gitlab Registry, jFrog Artifactory, AWS S3-Bucket, Sonatype Nexus, etc.)

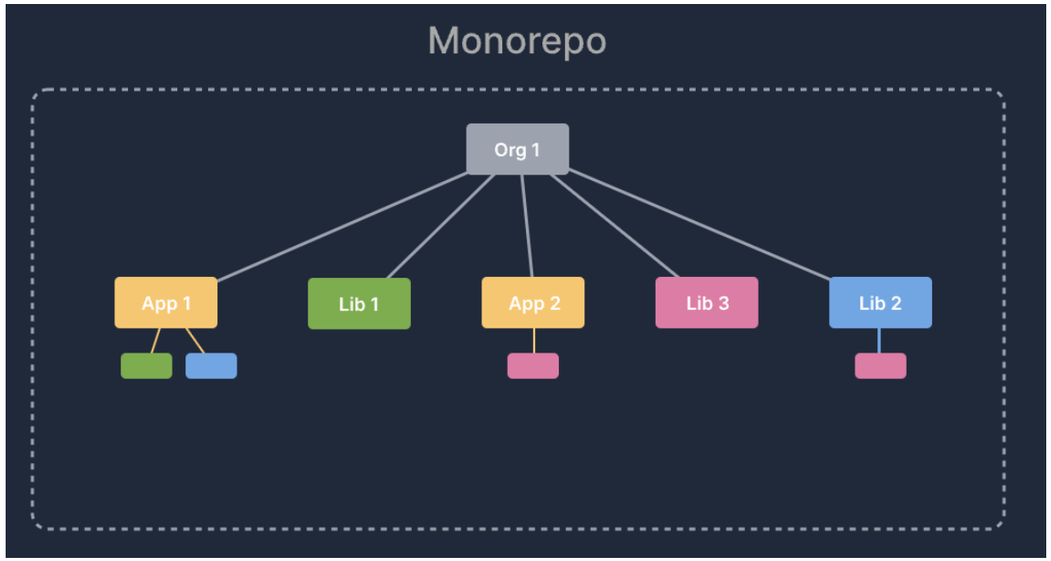
How to cut a repository?



One Codebase



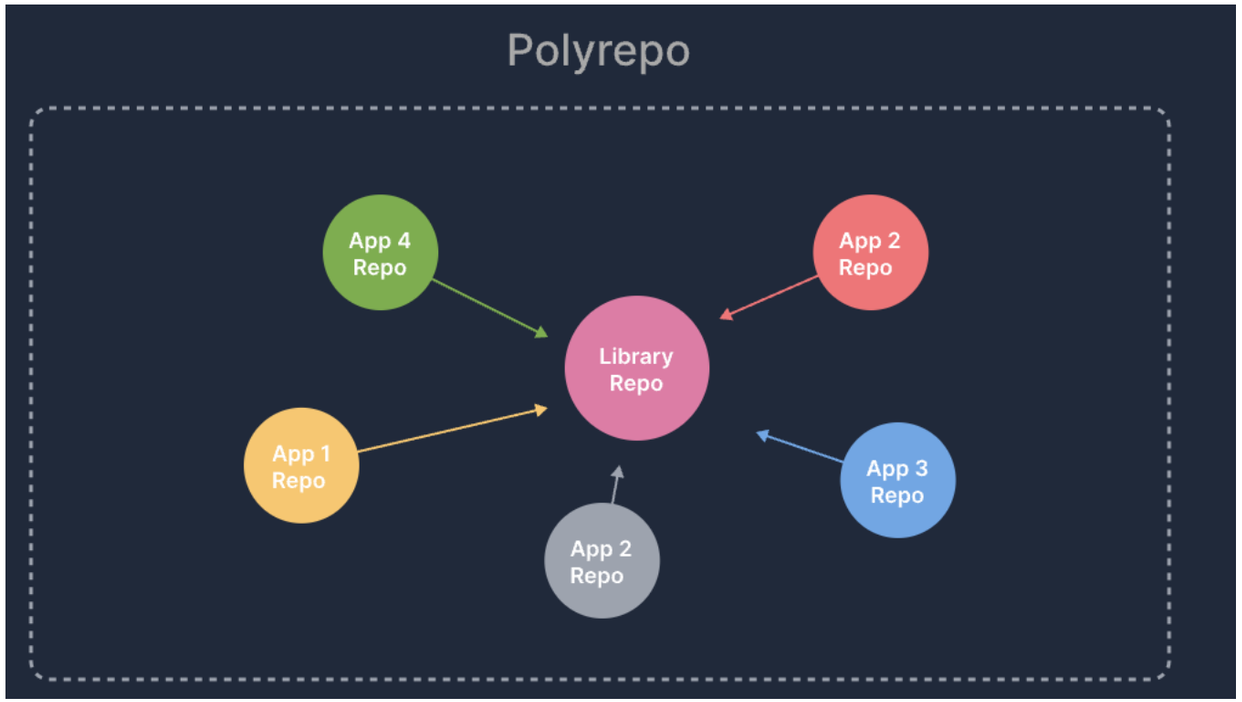
Monorepo VS Polyrepo



Monorepo:
Entire service architecture in one repository

- Self containing
- Sharing of common dependencies
- Easier, large scale refactorings

Monorepo VS Polyrepo



Polyrepo:
Each single service is in one repository

- Share of one microservice to other architectures
- Scaling of size of repository
- Security / Visibility / Atomicity is more natural
- Decoupling things that do not belong together

Sitenote: How Google managed sourcecode

How Google organizes Sourcecode:
https://www.youtube.com/watch?v=W71BTkUbdqE
https://cacm.acm.org/magazines/2016/7/204032-why-google-stores-billions-of-lines-of-code-in-a-single-repository/fulltext

Learning Git Branching-App
https://learngitbranching.js.org/

Linus on Git @ Google:
https://www.youtube.com/watch?v=idLyobOhtO4

Git - Conventional Commits

Structure Commits with a prefix

→ Conventional Commits

```
<type>[optional scope]: <description>
[optional body]
[optional footer(s)]
```

Type:

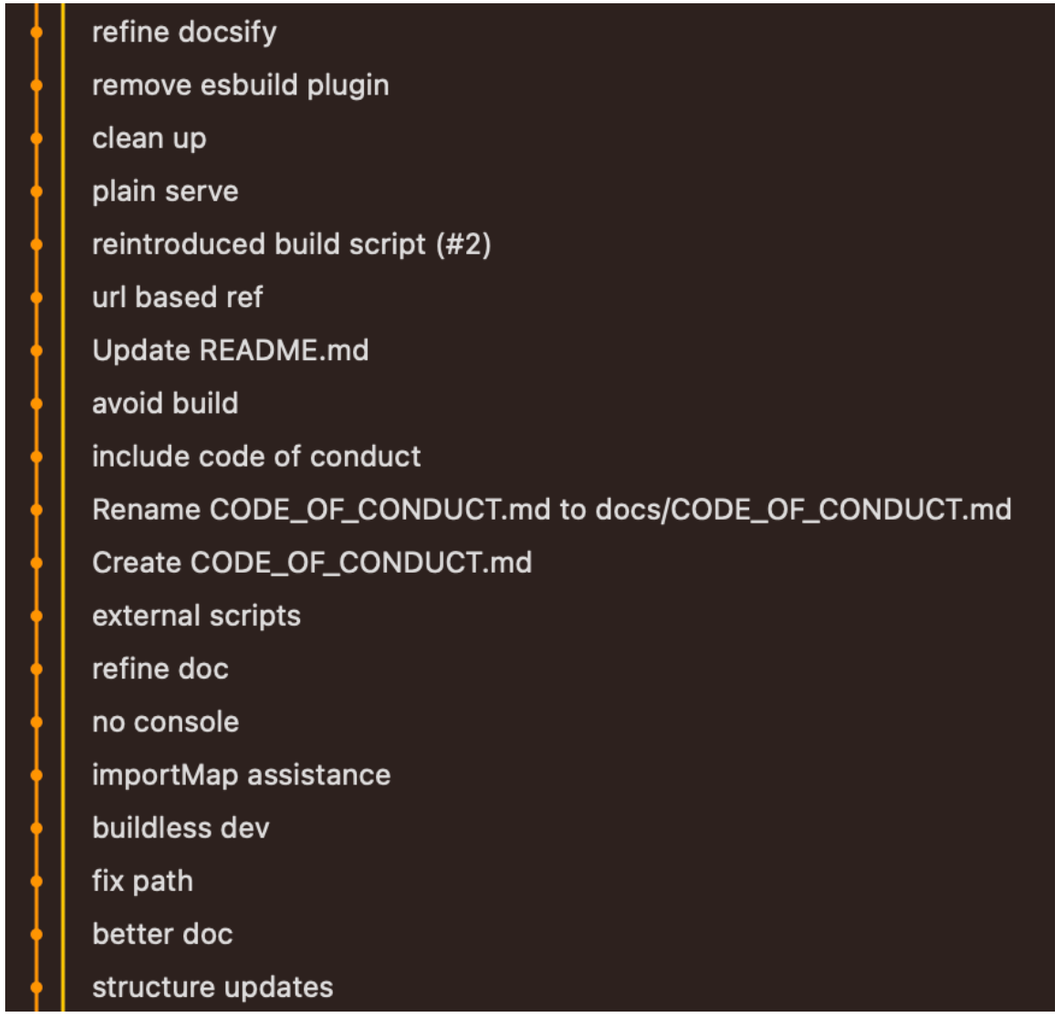
– feat, perf, chore, test

Scope:

- Part where this commit is applied

Optional Footers:

- (Meta)Information like `BREAKING CHANGE, Refs, Signed-by`



https://www.conventionalcommits.org/en/v1.0.0/#summary

Git – Advanced Handling

Conventional Commits are powerful:

- https://git-scm.com/docs/git-interpret-trailers
- `git log --pretty="%s"`

How to revert / rewrite commits?

- `git commit –amend`
- `git rebase –i`

Rebase VS Merging

- rebase: easier to rewrite/squashing, merge conflicts more complex
- Merge: easier to perform by conflicts, rewrites more complex

Gitlab – Handling

Reviews

- Merge Requests / Pull Requests
- Templating for Issues / Reference to git

Builds

- Can be branch/tag aware
- Can make use of branches / generated by automatisms

Force

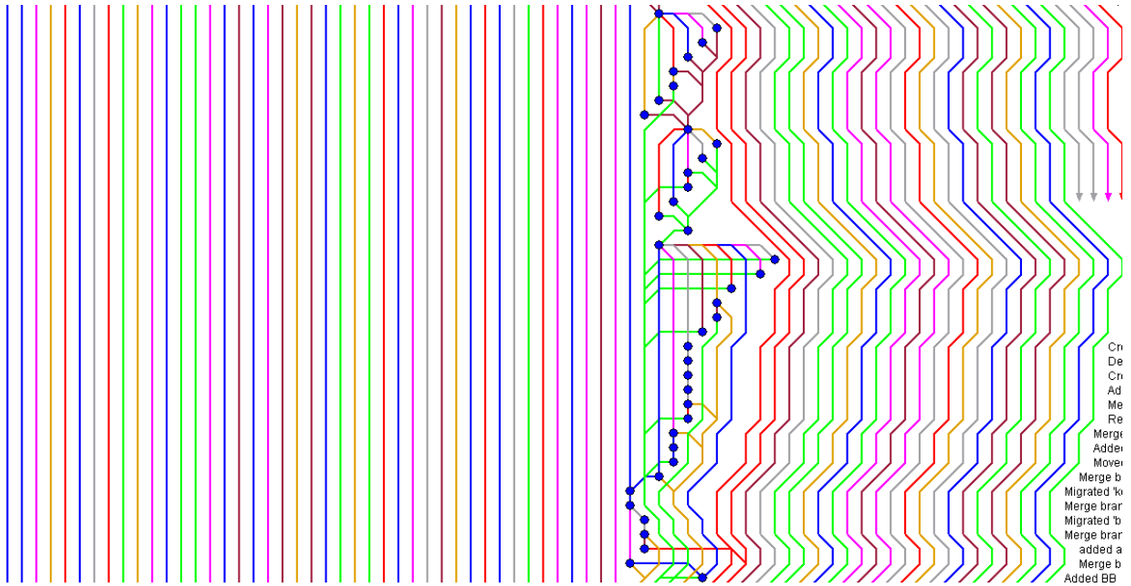
- rebase: results in rewritten history if already pushed. `-force` push necessary
- Getting wrong data out (large blobs, credentials) also needs `--force`

Branching Strategy

Git Branching is key benefit for handling source code:

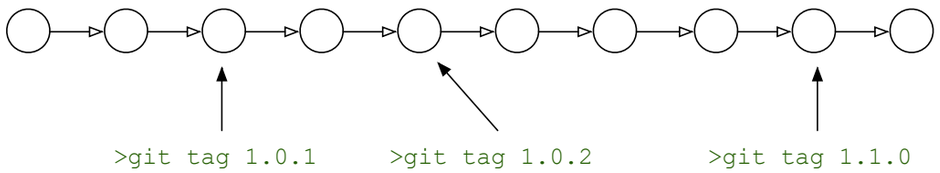
- Commits are either based on one predecessor or on two predecessors (merge)
- Branches (and Tags) are just labels for commits
  - Branches are cheap, easy and might by volatile

  → Inherent need for strategies how to handle branches



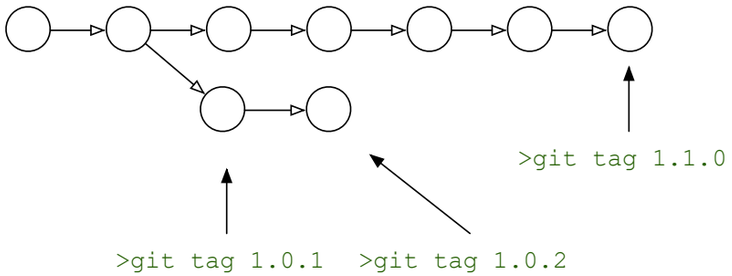
Branching Strategy

Trunk based



- High cadence of releasing
- Fast forward fixing
- Short term branches

Gitflow Branching



- Long term branches for delivery
- Short term branches for features

https://trunkbaseddevelopment.com/release-from-trunk/
https://learning.oreilly.com/library/view/ci-cd-design-patterns/9781835889640/B22089_05.xhtml#_idParaDest-94

Releasing

SemVer:
– MAJOR version increments indicate incompatible API changes,
– MINOR version increments add functionality in a backward-compatible manner,
– PATCH version increments make backward-compatible bug fixes.

CalVer:
- YYYY.MM.DD: Full date (e.g., 2021.03.22) indicating the exact release day.
- YYYY.MM: Year and month (e.g., 2021.03) for monthly releases.

ZeroVer:
- Versions might look like 0.1.0, 0.2.0, and so on.
- Never make it to 1.0.0

HashVer / GithashVer:
- Full year (printf("%Y"))
- Zero padded month (printf("%m"))
- [Optional] Zero padded day (printf("%d"))
- 10+ characters of the current source control commit's hash

https://github.com/andrew/nesbitt.io/blob/master/_posts/2024-06-24-from-zerover-to-semver-a-comprehensive-list-of-versioning-schemes-in-open-source.md

What is a release?

Maven

- Identified by fixed version in GAV(C)
- Unique with respect to GAV(C)
- Not to be modified again
  - Seen as “stable”
  - Pro`tected for overriding artifact

Otherwise, transitive dependencies will break

For continuous development,
please use “-SNAPSHOT”-suffix

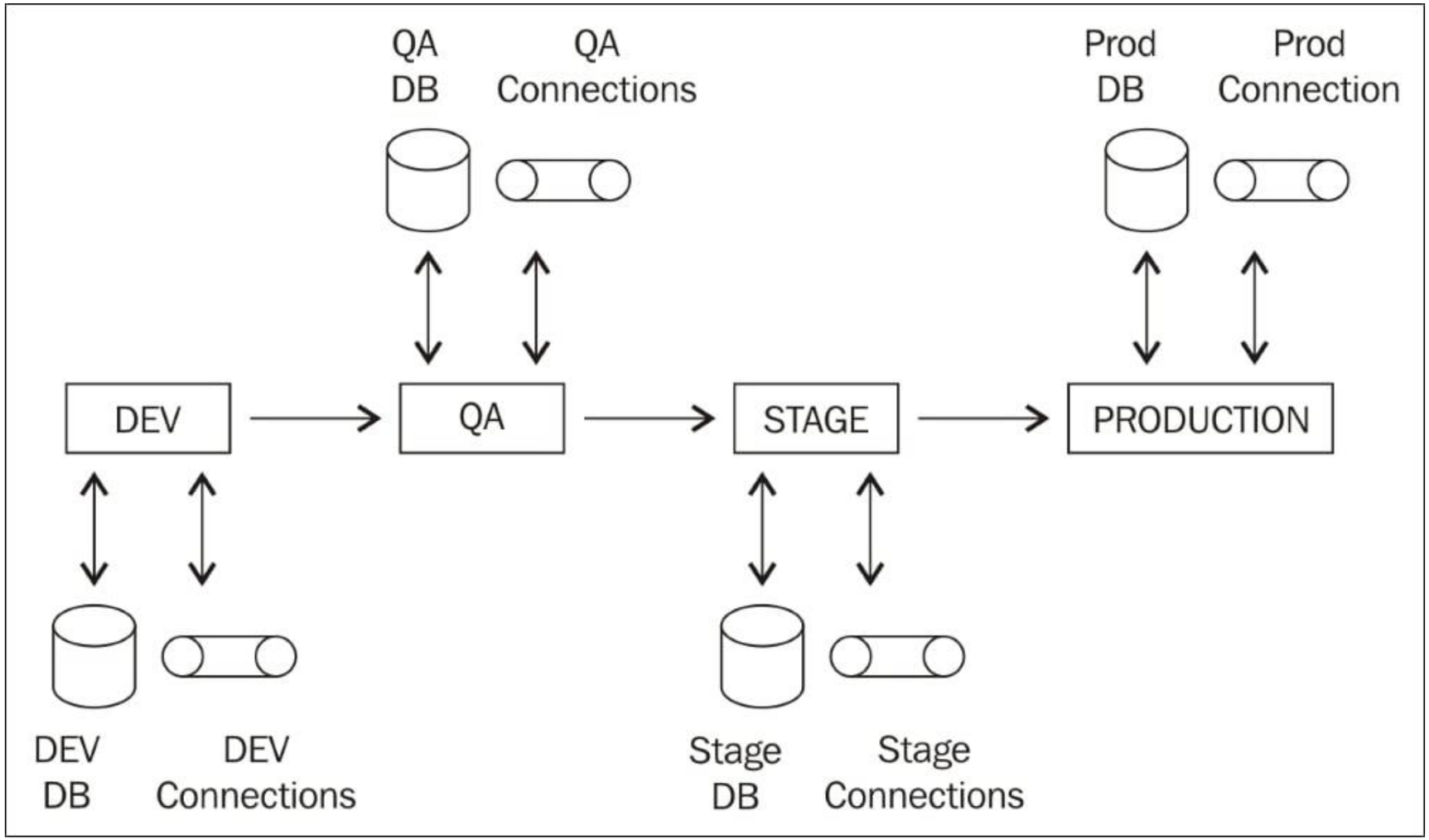
Docker / OCI

- Identified by Tags, Hash
- Unique with respect of NAME:TAG@HASH
- NAME:TAG can be overridden

- No danger of breaking transitive dependencies -> Images are self-contained

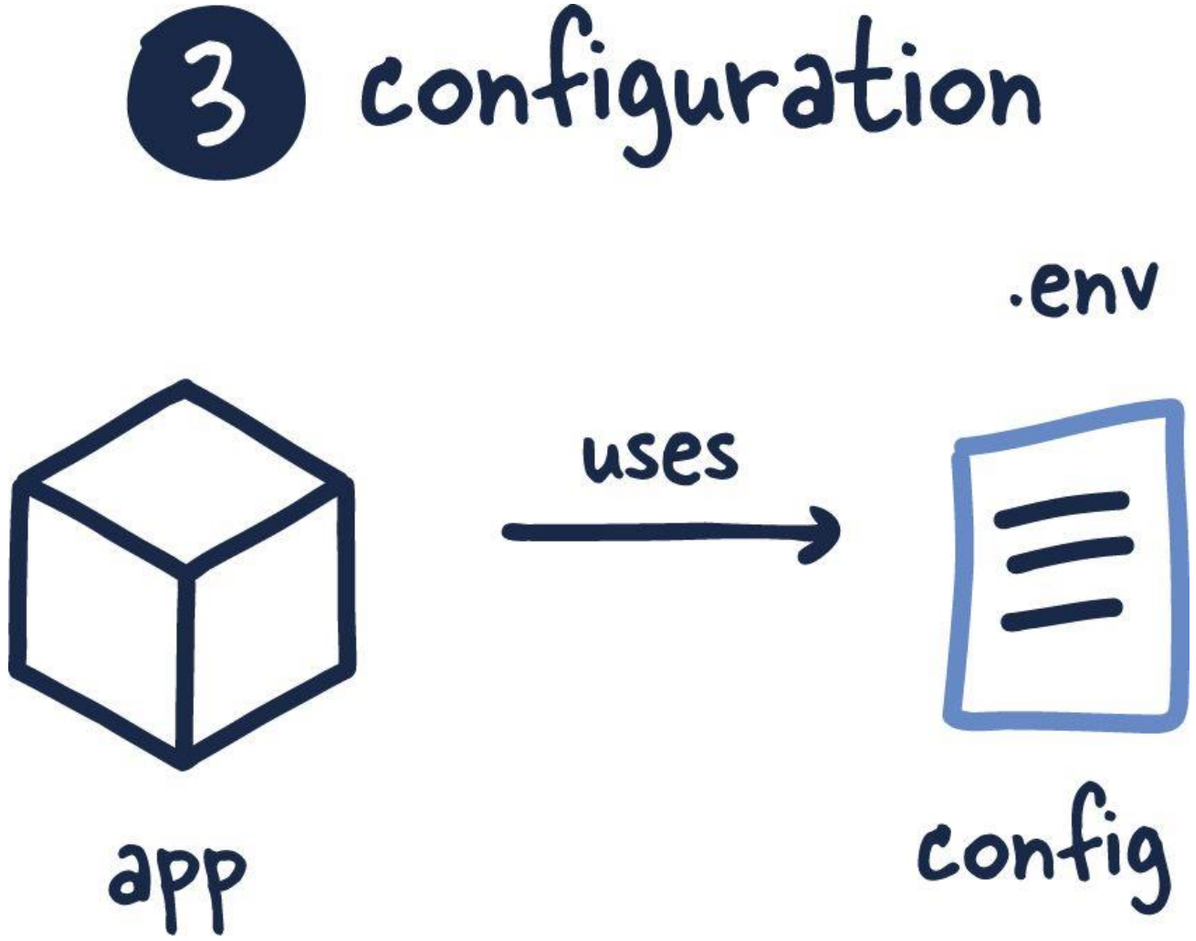
For continuous development,
please use the tag “latest”

Environment configurations separate from application



Environment configurations separate from application

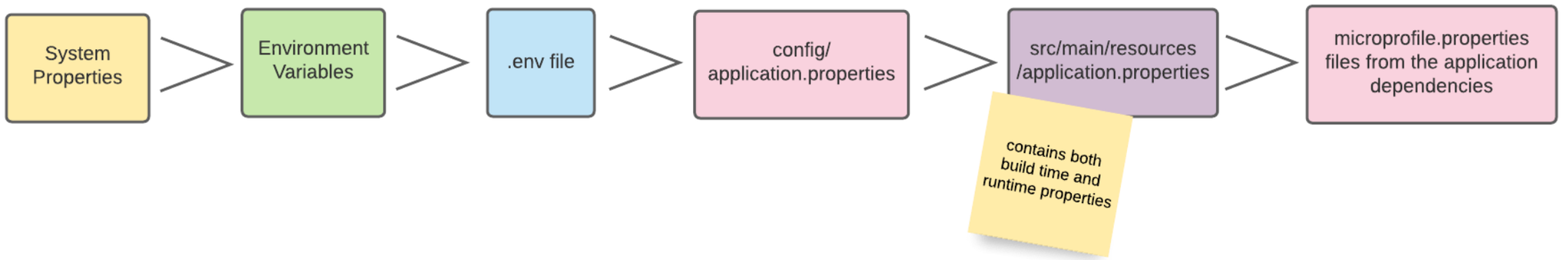
- Apps sometimes store config as constants in the code. This is a violation of twelve-factor, which requires **strict separation of config from code**. Config varies substantially across deploys, code does not.[…]
- **The twelve-factor app stores config in *environment variables*** (often shortened to *env vars* or *env*)



Quarkus Example

```java
public class PersistenceModule {
    no usages  ginccc +1
    @Produces
    @ApplicationScoped
    public MongoDatabase provideMongoDB(@ConfigProperty(name = "mongodb.connectionString") String connectionString,
                                        @ConfigProperty(name = "mongodb.database") String database) {
```

Order of configurations



Where to inject which configurations?

Use fixed defined locations for configurations

<u>“Stable” configurations</u>
Standard Configuration to run locally

-> Put them near the Compile-Time
e.g. application.properties in Jar

<u>“Runtime” configurations</u>
Stage-dependent configurations

-> Put them near the Runtime
e.g. System-variables / .env-File

slido

Please download and install the Slido app on all computers you use

How do you handle logger configuration?

ⓘ Start presenting to display the poll results on this slide.

slido

Please download and install the Slido app on all computers you use

What kind is the configuration of an external database?

ⓘ Start presenting to display the poll results on this slide.

slido

Please download and install the Slido app on all computers you use

What kind is the endpoint for your monitoring?

ⓘ Start presenting to display the poll results on this slide.

slido

Please download and install the Slido app on all computers you use

How to you handle the binding to an authentification service?

ⓘ Start presenting to display the poll results on this slide.

slido

Please download and install the Slido app on all computers you use

What kind of configuration is the setting of an external REST-Service?

ⓘ Start presenting to display the poll results on this slide.

slido

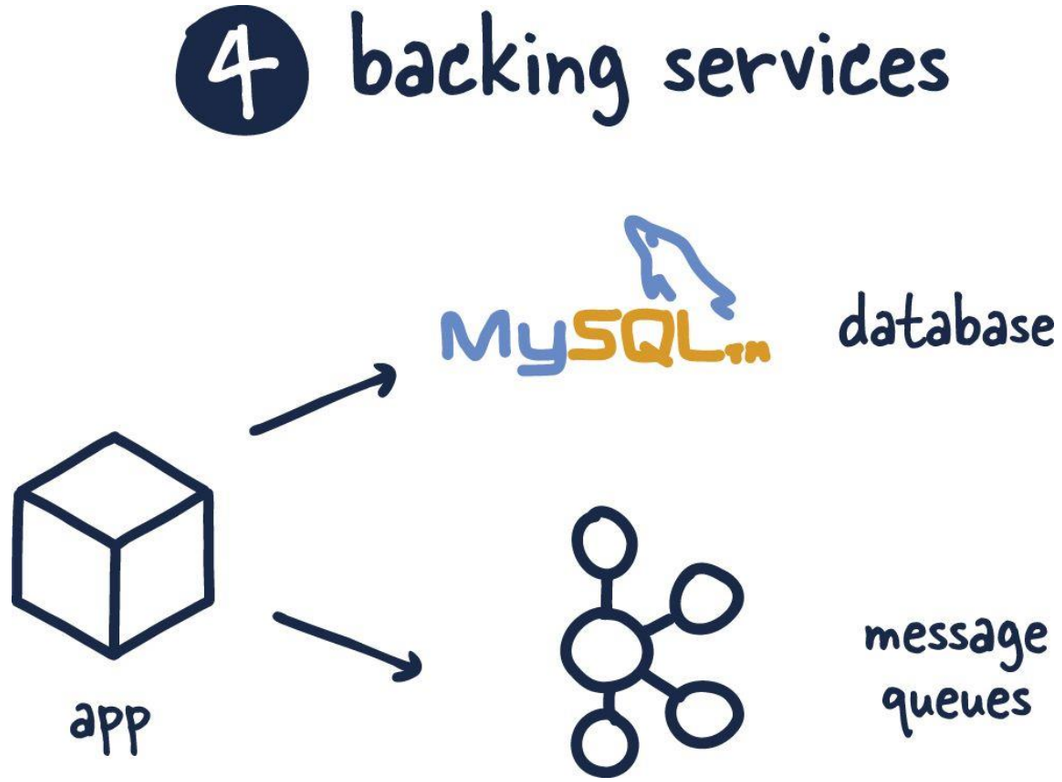
Please download and install the Slido app on all computers you use

Where do you set the plain application name?

ⓘ Start presenting to display the poll results on this slide.

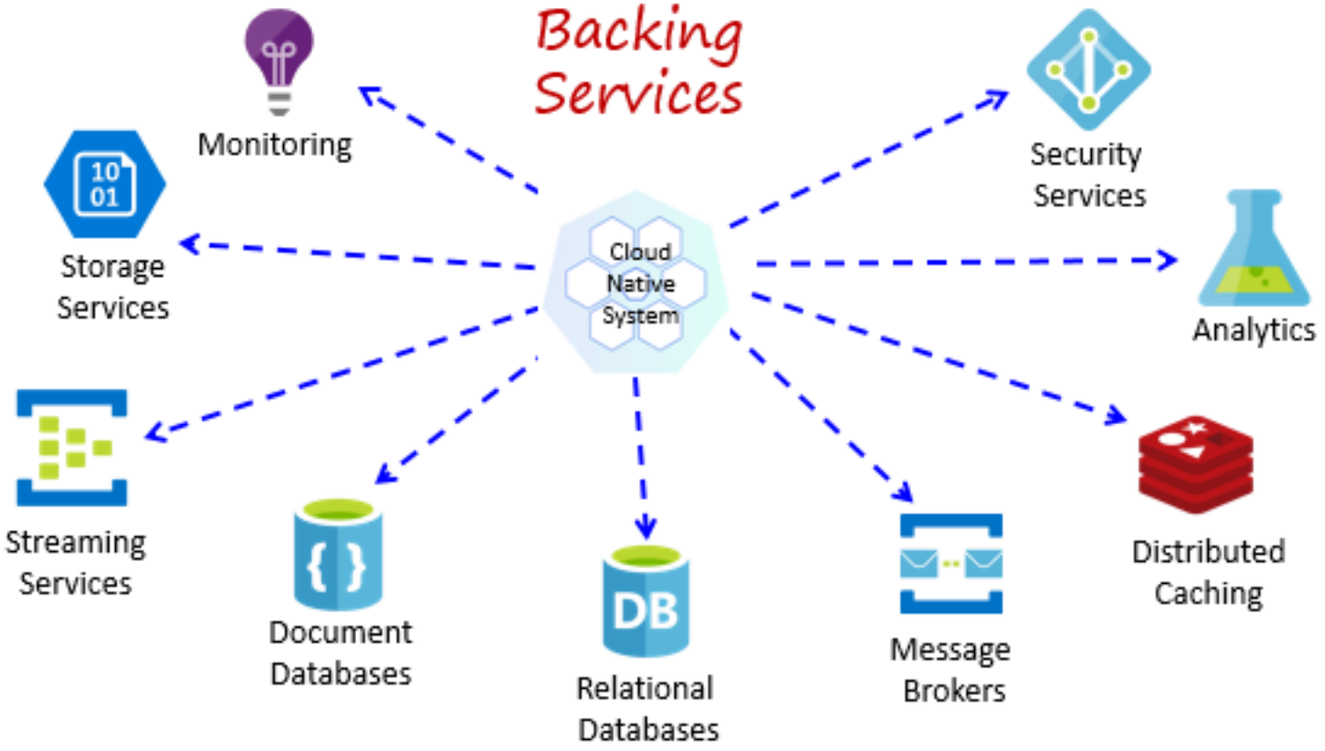
Backing Services

- A *backing service* is any service the app consumes over the network as part of its normal operation. […]
- **The code for a twelve-factor app makes no distinction between local and third party services.** To the app, both are attached resources, accessed via a URL or other locator/credentials stored in the config.



Backing Services

- Externalize Services where applicable
  - Storage
  - Databases
  - Messaging
  - Identity Services
  - Monitoring & Logging

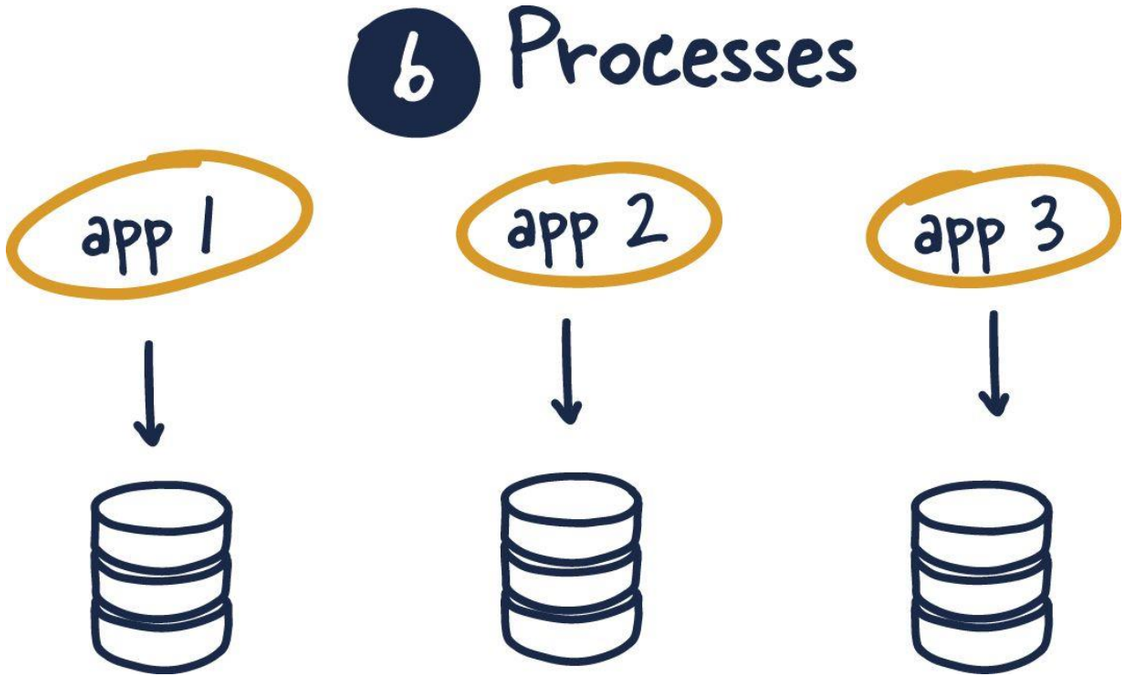


Externalize your pain
- Storage is hard on scale
- Security is hard in an organization
- Inter-Application communication is complicated

Focus on business code you can develop and operate

Execute the app as stateless process

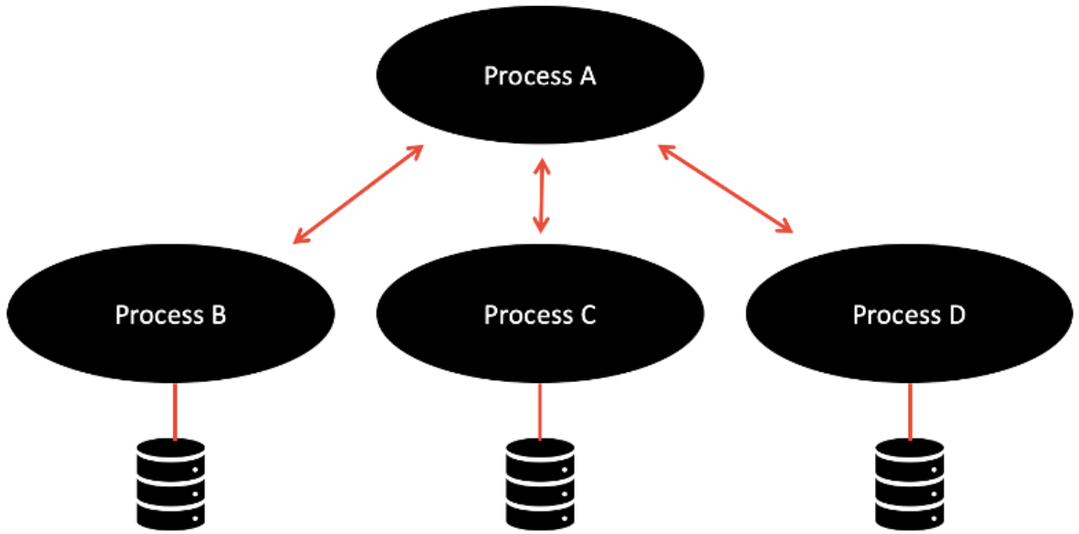
- The app is executed in the execution environment as one or more *processes*. […]
- **Twelve-factor processes are stateless and share-nothing.** […]
- The memory space or filesystem of the process can be used as a brief, single-transaction cache. For example, downloading a large file, operating on it, and storing the results of the operation in the database. The twelve-factor app never assumes that anything cached in memory or on disk will be available on a future request or job.
-



Execute the app as stateless process

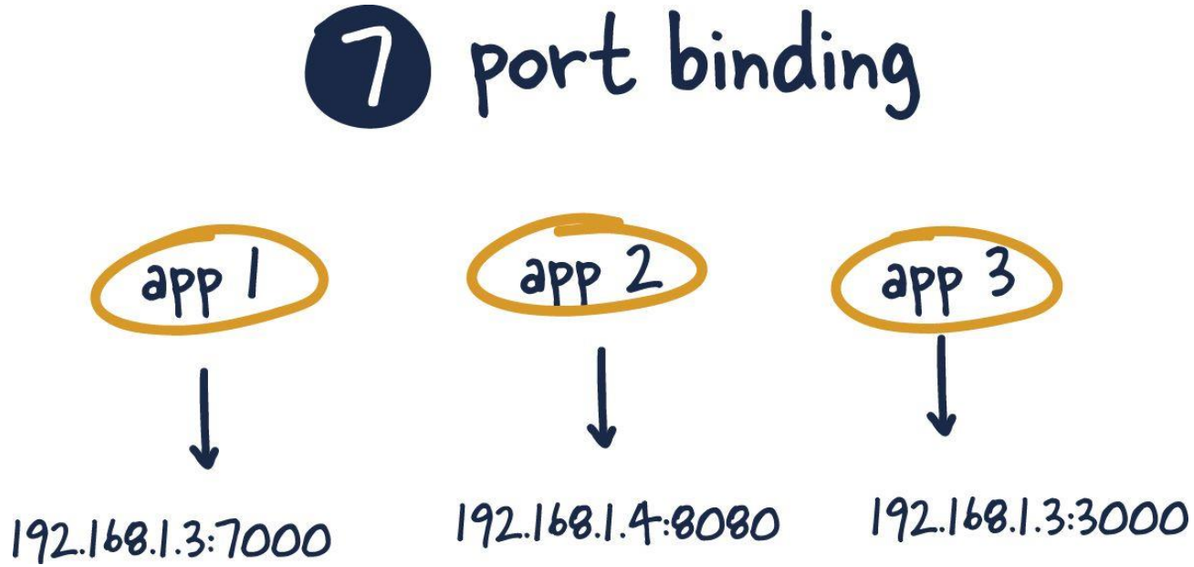
Processes share **nothing:**

- State must be stored externally
  - Database
  - Cache
- Isolated from other processes, future starts
  - Cleanup of local resources before/after starts



Communicate via fixed ports

- **The twelve-factor app is completely self-contained** and does not rely on runtime injection of a webserver into the execution environment to create a web-facing service. The web app **exports HTTP as a service by binding to a port**, and listening to requests coming in on that port. […]
- Note also that the port-binding approach means that one app can become the backing service for another app, by providing the URL to the backing app as a resource handle in the config for the consuming app.

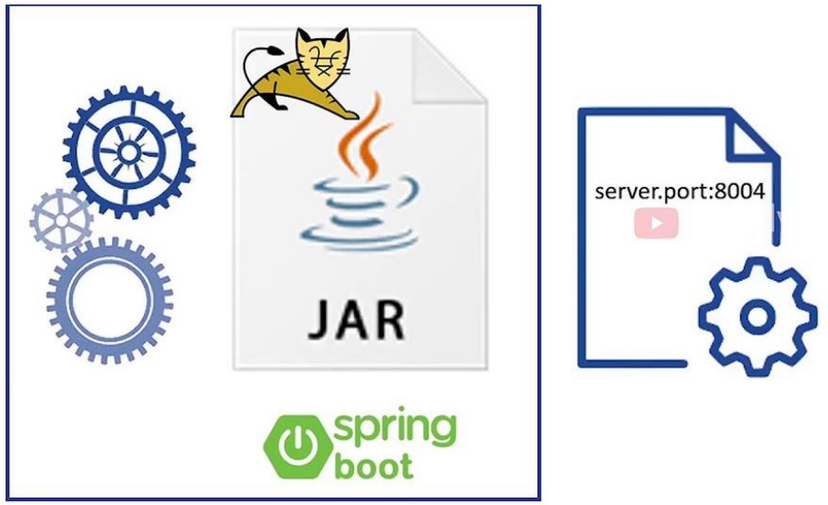


Define one port for each service.

Communicate via fixed ports

– Each application / service has fixed port(s).

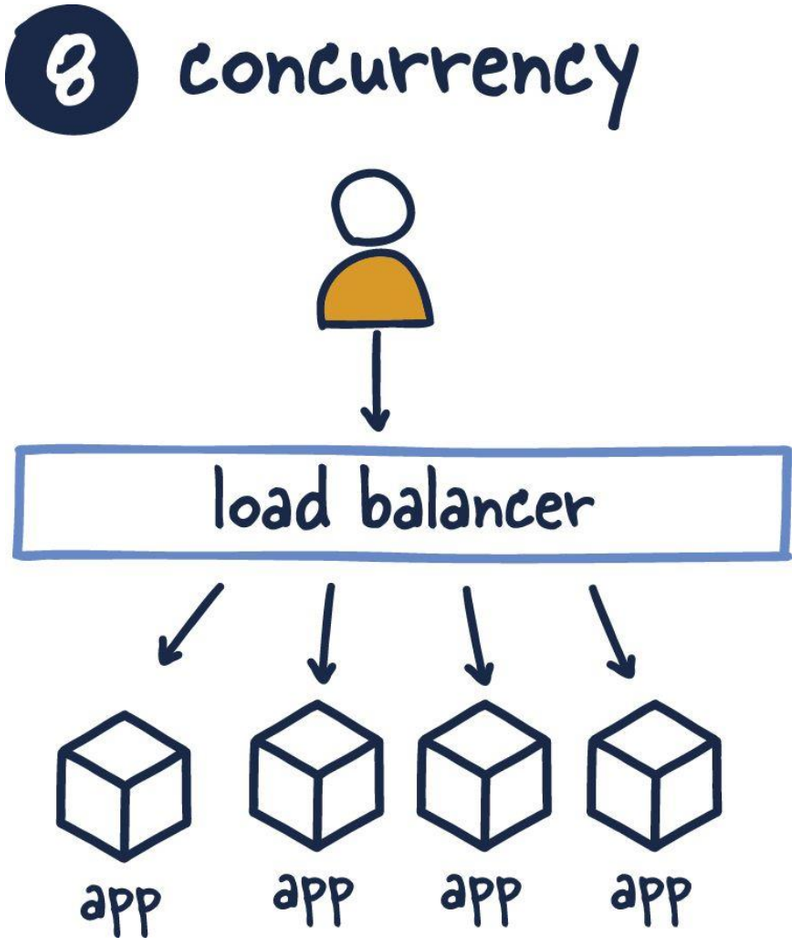
- Local: Beware of port collisions
- Remote: Port Collisions can be handled by the platform
- No other access to the service should be possible.



Choose different ports and rely on them.

Use process model for scalability

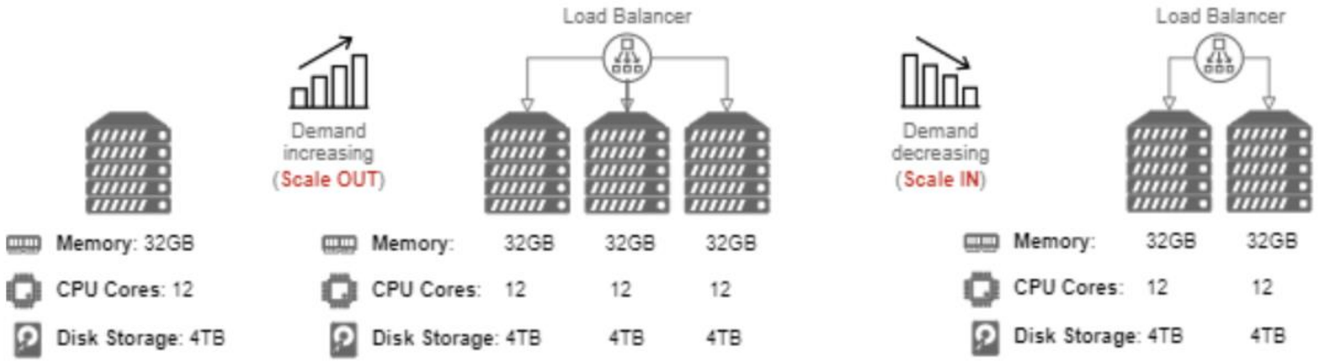
- **In the twelve-factor app, processes are a first class citizen.** Processes in the twelve-factor app take strong cues from the unix process model for running service daemons. […]
- But an individual VM can only grow so large (vertical scale), so the application must also be able to span multiple processes running on multiple physical machines.



Use process model for scalability

– Horizontal Scaling comes free of charge:

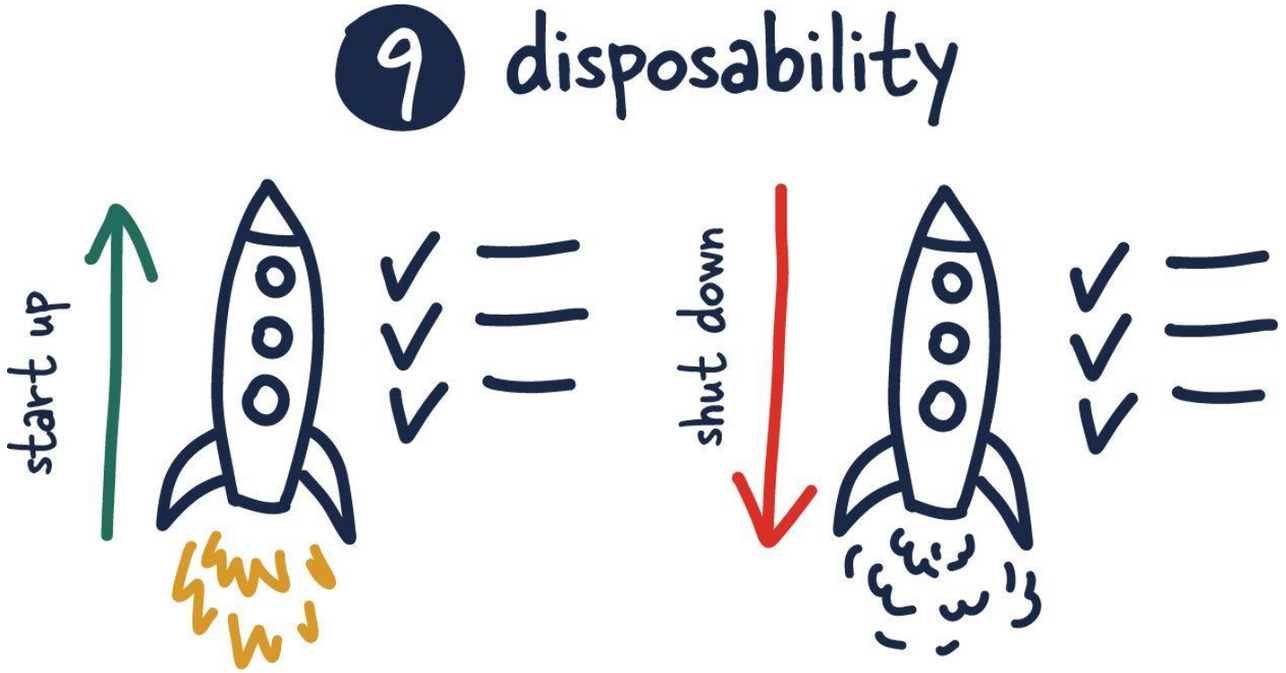
- No local state (see 6.)
- No shared state (see 7.)
- Relying on process management of system

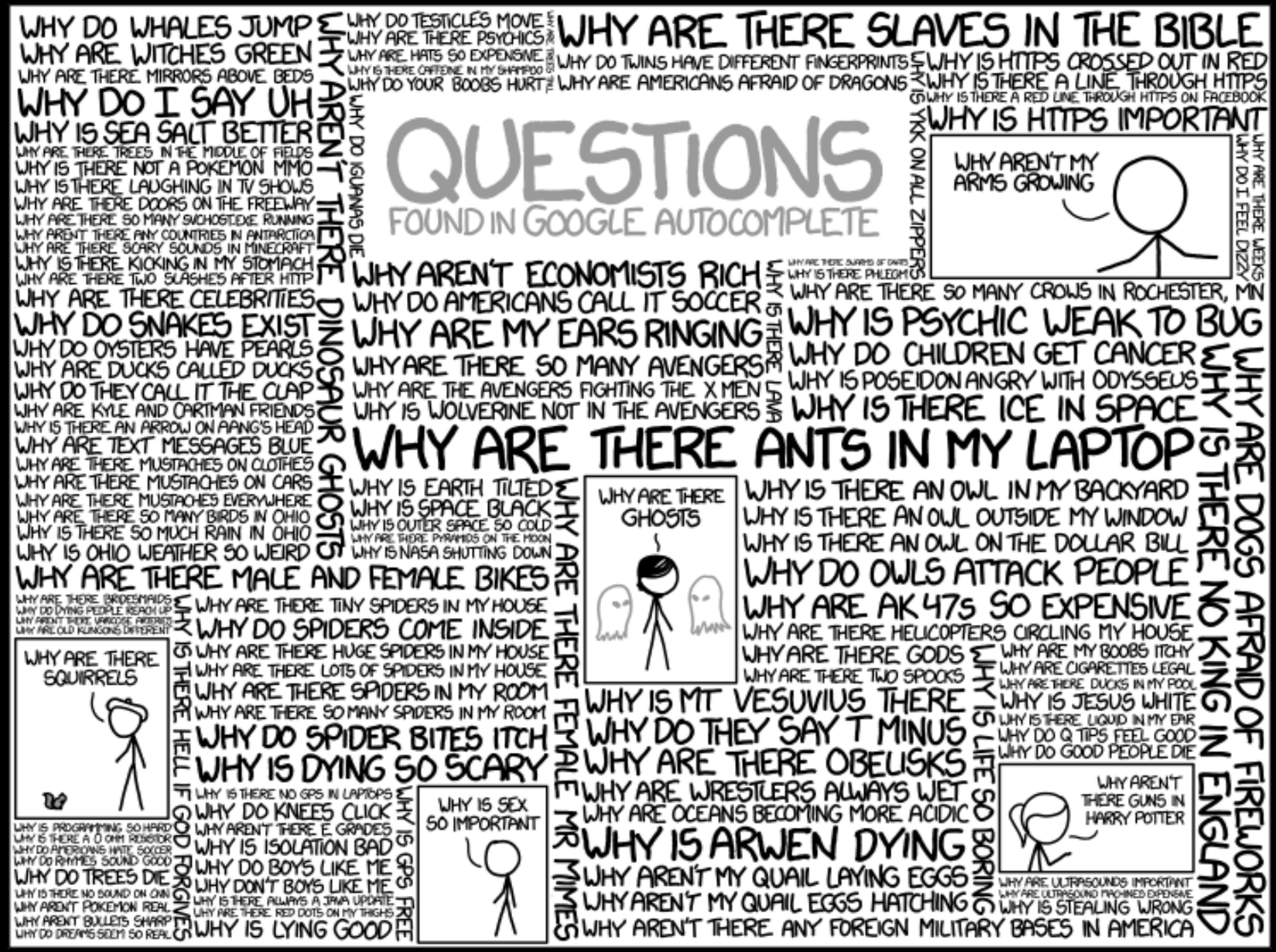


– Scalability is guaranteed by combination of criteria so far.

Start fast and shutdown graceful

- **The twelve-factor app's processes are *disposable*, meaning they can be started or stopped at a moment's notice.** This facilitates fast elastic scaling, rapid deployment of code or config changes, and robustness of production deploys. […]
- Processes should strive to **minimize startup time**. […]
- Processes **shut down gracefully when they receive a SIGTERM** signal from the process manager.




QUESTIONS
FOUND IN GOOGLE AUTOCOMPLETE
WHY DO WHALES JUMP
WHY ARE WITCHES GREEN
WHY ARE THERE MIRRORS ABOVE BEDS
WHY DO I SAY UH
WHY IS SEA SALT BETTER
WHY ARE THERE TREES IN THE MIDDLE OF FIELDS
WHY IS THERE NOT A POKEMON MMO
WHY IS THERE LAUGHING IN TV SHOWS
WHY ARE THERE DOORS ON THE FREEWAY
WHY ARE THERE SO MANY SYCHOSTICKS RUNNING
WHY AREN'T THERE ANY COUNTRIES IN ANTARCTICA
WHY ARE THERE SCARY SOUNDS IN MINECRAFT
WHY IS THERE KICKING IN MY STOMACH
WHY ARE THERE TWO SLASHES AFTER HTTP
WHY ARE THERE CELEBRITIES
WHY DO SNAKES EXIST
WHY DO OYSTERS HAVE PEARLS
WHY ARE DUCKS CALLED DUCKS
WHY DO THEY CALL IT THE CLAP
WHY ARE KYLE AND CARTMAN FRIENDS
WHY IS THERE AN ARROW ON AANG'S HEAD
WHY ARE TEXT MESSAGES BLUE
WHY ARE THERE MUSTACHES ON CLOTHES
WHY ARE THERE MUSTACHES ON CARS
WHY ARE THERE MUSTACHES EVERYWHERE
WHY ARE THERE SO MANY BIRDS IN OHIO
WHY IS THERE SO MUCH RAIN IN OHIO
WHY IS OHIO WEATHER SO WEIRD
WHY ARE THERE MALE AND FEMALE BIKES
WHY ARE THERE BRIDESMAIDS
WHY DO DYING PEOPLE REARM UP
WHY AREN'T THERE VARICOSE AIRBIKES
WHY ARE OLD KLINGONS DIFFERENT
WHY ARE THERE SQUIRRELS
WHY IS PROGRAMMING SO HARD
WHY IS THERE A 0 OHM RESISTOR
WHY DO AMERICANS HATE SOCCER
WHY DO RHYMES SOUND GOOD
WHY DO TREES DIE
WHY IS THERE NO SOUND ON CNN
WHY AREN'T POKEMON REAL
WHY AREN'T BULLETS SHARP
WHY DO DREAMS SEEM SO REAL
WHY ARE THERE TINY SPIDERS IN MY HOUSE
WHY DO SPIDERS COME INSIDE
WHY ARE THERE HUGE SPIDERS IN MY HOUSE
WHY ARE THERE LOTS OF SPIDERS IN MY HOUSE
WHY ARE THERE SPIDERS IN MY ROOM
WHY ARE THERE SO MANY SPIDERS IN MY ROOM
WHY DO SPIDER BITES ITCH
WHY IS DYING SO SCARY
WHY IS THERE NO GPS IN LAPTOPS
WHY DO KNEES CLICK
WHY AREN'T THERE E GRADES
WHY IS ISOLATION BAD
WHY DO BOYS LIKE ME
WHY DON'T BOYS LIKE ME
WHY IS THERE ALWAYS A JAVA UPDATE
WHY ARE THERE RED DOTS ON MY THIGHS
WHY IS LYING GOOD
WHY DO TESTICLES MOVE
WHY ARE THERE PSYCHICS
WHY ARE HATS SO EXPENSIVE
WHY IS THERE CAFFEINE IN MY SHAMPOO
WHY DO YOUR BOOBS HURT
WHY ARE THERE SLAVES IN THE BIBLE
WHY DO TWINS HAVE DIFFERENT FINGERPRINTS
WHY ARE AMERICANS AFRAID OF DRAGONS
WHY IS HTTPS CROSSED OUT IN RED
WHY IS THERE A LINE THROUGH HTTPS
WHY IS THERE A RED LINE THROUGH HTTPS ON FACEBOOK
WHY IS HTTPS IMPORTANT
WHY AREN'T MY ARMS GROWING
WHY IS YKK ON ALL ZIPPERS
WHY ARE THERE WEEKS
WHY DO I FEEL DIZZY
WHY DO IGUANAS DIE
WHY AREN'T ECONOMISTS RICH
WHY DO AMERICANS CALL IT SOCCER
WHY ARE MY EARS RINGING
WHY ARE THERE SO MANY AVENGERS
WHY ARE THE AVENGERS FIGHTING THE X MEN
WHY IS WOLVERINE NOT IN THE AVENGERS
WHY ARE THERE SWARMS OF GNATS
WHY IS THERE PHLEGM
WHY ARE THERE SO MANY CROWS IN ROCHESTER, MN
WHY IS PSYCHIC WEAK TO BUG
WHY DO CHILDREN GET CANCER
WHY IS POSEIDON ANGRY WITH ODYSSEUS
WHY IS THERE ICE IN SPACE
WHY IS THERE LAVA
WHY ARE THERE ANTS IN MY LAPTOP
WHY IS EARTH TILTED
WHY IS SPACE BLACK
WHY IS OUTER SPACE SO COLD
WHY ARE THERE PYRAMIDS ON THE MOON
WHY IS NASA SHUTTING DOWN
WHY ARE THERE GHOSTS
WHY IS THERE AN OWL IN MY BACKYARD
WHY IS THERE AN OWL OUTSIDE MY WINDOW
WHY IS THERE AN OWL ON THE DOLLAR BILL
WHY DO OWLS ATTACK PEOPLE
WHY ARE AK 47s SO EXPENSIVE
WHY ARE THERE HELICOPTERS CIRCLING MY HOUSE
WHY ARE THERE GODS
WHY ARE THERE TWO SPOOKS
WHY IS MT VESUVIUS THERE
WHY DO THEY SAY T MINUS
WHY ARE THERE OBELISKS
WHY ARE WRESTLERS ALWAYS WET
WHY ARE OCEANS BECOMING MORE ACIDIC
WHY IS ARWEN DYING
WHY AREN'T MY QUAIL LAYING EGGS
WHY AREN'T MY QUAIL EGGS HATCHING
WHY AREN'T THERE ANY FOREIGN MILITARY BASES IN AMERICA
WHY IS SEX SO IMPORTANT
WHY ARE MY BOOBS ITCHY
WHY ARE CIGARETTES LEGAL
WHY ARE THERE DUCKS IN MY POOL
WHY IS JESUS WHITE
WHY IS THERE LIQUID IN MY EAR
WHY DO Q TIPS FEEL GOOD
WHY DO GOOD PEOPLE DIE
WHY AREN'T THERE GUNS IN HARRY POTTER
WHY ARE ULTRASOUNDS IMPORTANT
WHY ARE ULTRASOUND MACHINES EXPENSIVE
WHY IS STEALING WRONG
WHY ARE DOGS AFRAID OF FIREWORKS
WHY IS THERE NO KING IN ENGLAND
WHY IS LIFE SO BORING
WHY IS GPS FREE
WHY ARE THERE FEMALE MR MIMES
WHY IS THERE HELL IF GOD FORGIVES
WHY AREN'T THERE DINOSAUR GHOSTS