



Göttliche Regeln
Build your PaaS with
deis.io

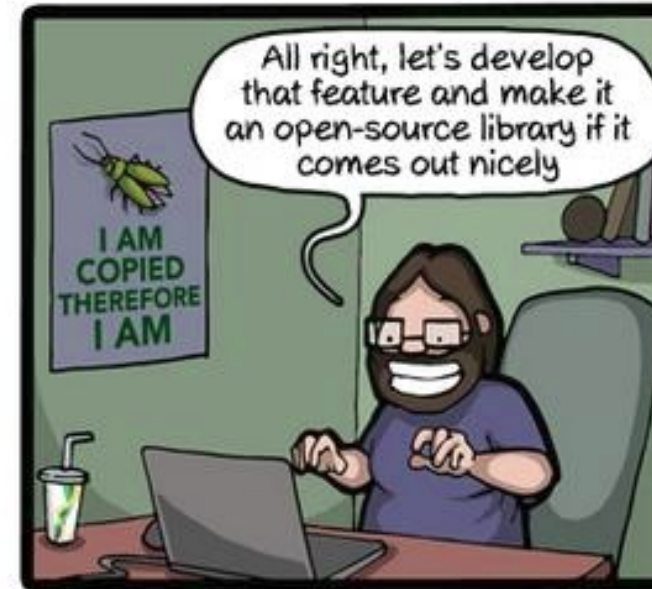
Nicolas Byl, codecentric AG





<http://dilbert.com/strip/2011-01-07>

Two ways of managing a new feature's development



CommitStrip.com

<http://www.commitstrip.com/en/2014/09/30/two-ways-of-managing-a-new-features-development/>

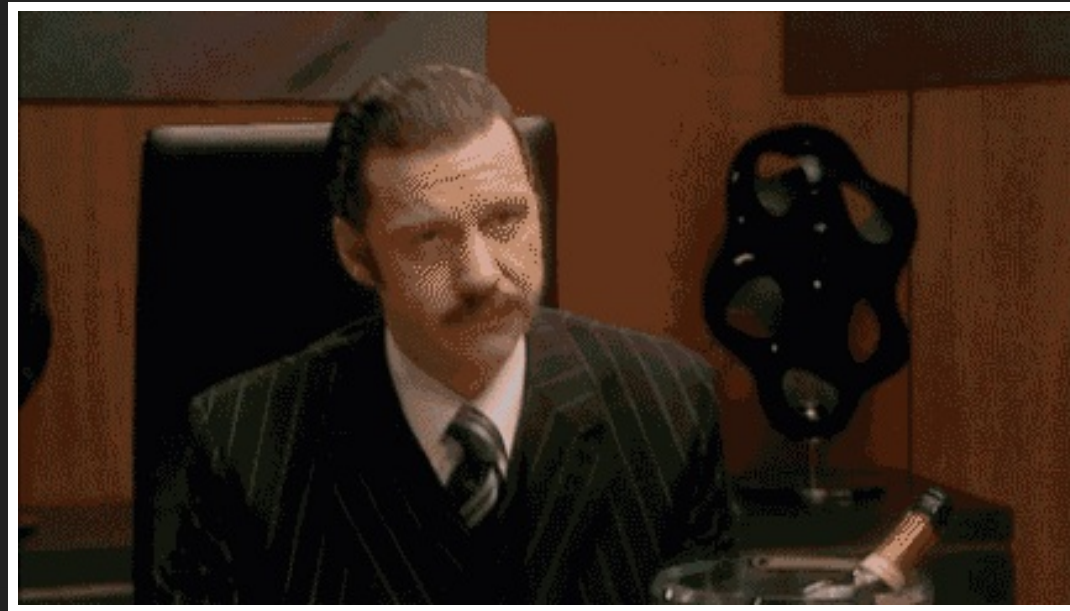
Platform as a Service



Docker



Operations and Docker



Docker

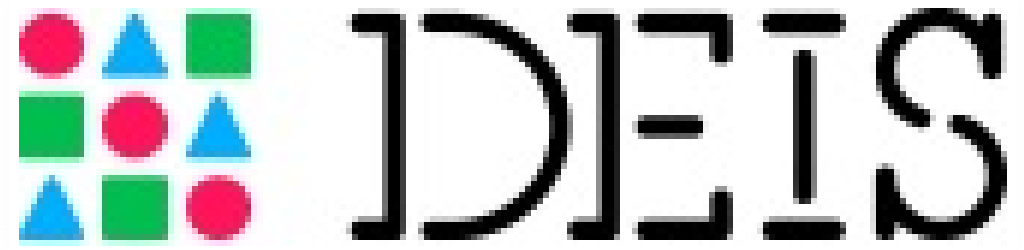
- Docker Container als Deliverable?
- Compliance?
- Application Lifecycle?

Twelve-Factor Applications



- **Codebase**
One codebase tracked in revision control, many deploys
- **Dependencies**
Explicitly declare and isolate dependencies
- **Config**
Store config in the environment
- **Backing Services**
Treat backing services as attached resources
- **Build, release, run**
Strictly separate build and run stages
- **Processes**
Execute the app as one or more stateless processes

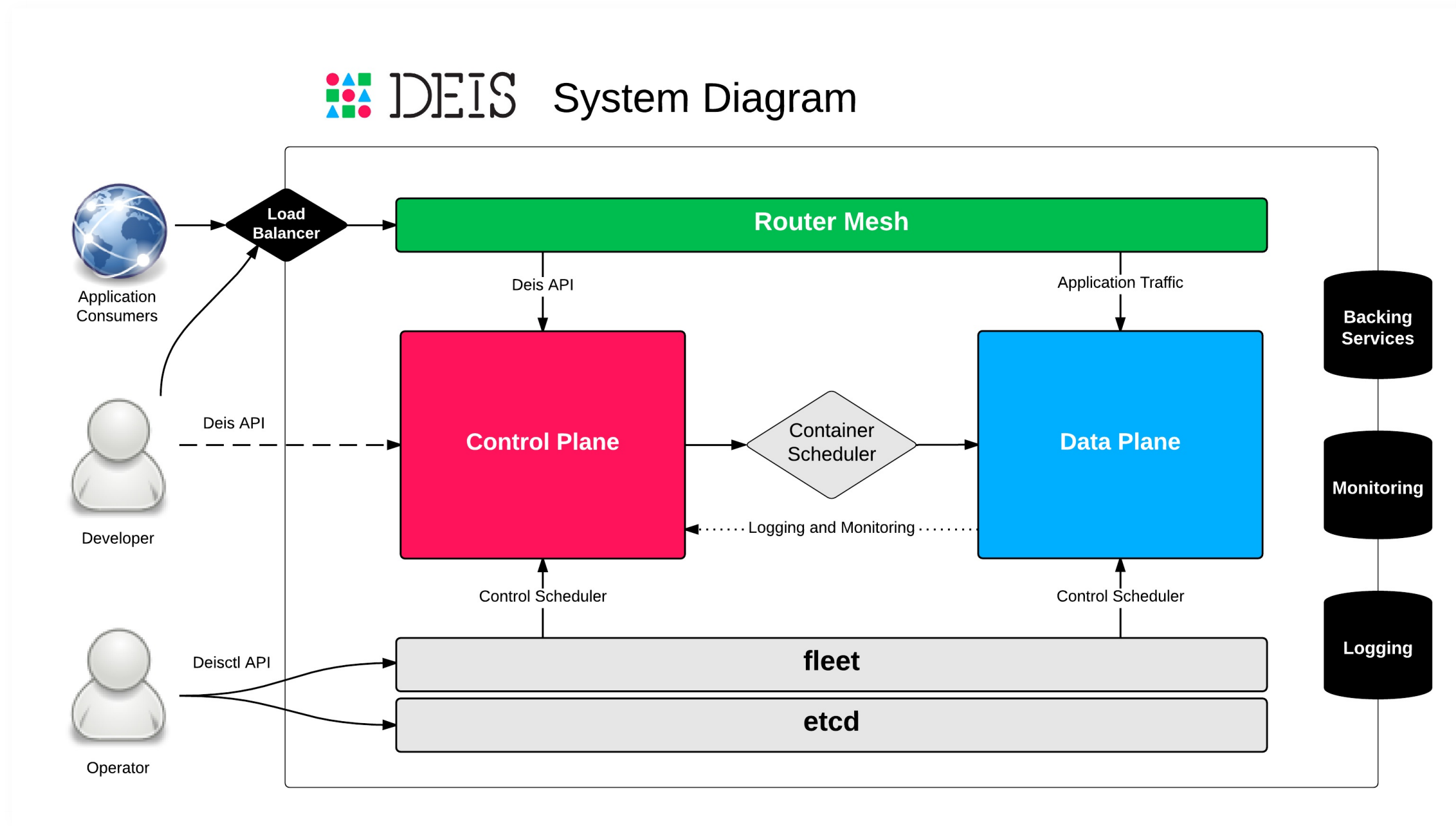
- **Port binding**
Export services via port binding
- **Concurrency**
Scale out via the process model
- **Disposability**
Maximize robustness with fast startup and graceful shutdown
- **Dev/prod parity**
Keep development, staging, and production as similar as possible
- **Logs**
Treat logs as event streams
- **Admin processes**
Run admin/management tasks as one-off processes



DEIS



Architektur



Deployment

```
git push deis master
```

Build Types

- Heroku Buildpacks
- Dockerfile
- Docker Images

Demo Time!

Ausblick

- Kubernetes (TP)
- Mesos / Marathon (TP)
- Docker Swarm (TP)
- Upgrades

DANKE FÜR DIE AUFMERKSAMKEIT

- <http://deis.io>
- <http://github.com/nbyl/deis-workshop>

Fragen? Antworten!

- nicolas.byl@codecentric.de
- [@NicolasByl](#)