



timetracker.io

PRODYNA Architecture Certification

1.0

## Lösungsvorstellung

Leonid Agranovskiy

26.06.2015 Eschborn

# ➤ Agenda:

1. Unser Versprechen
2. Unsere Lösung
3. Unsere Prozesse
4. Unsere Qualität
5. Unser Monitoring
6. Unsere Demo

# ➤ Unser Versprechen

## Das haben Sie bestellt

### Projektverwaltung

- Übersicht, Erfassung und Bearbeitung
- Mitarbeiterzuordnung



### Buchungsverwaltung

- Zeitenerfassen
- Projektbezug



### Benutzerverwaltung

- Selbstregistrierung
- Ändern, Kennwort-Reset
- Deaktivieren von Benutzer



### Smart Refresh

- Wenn sich Sichten ändern – Refresh ohne Neuladen



### Kollision Freiheit

- Buchungen von einem MA dürfen nicht mit einander kollidieren



### Performance

- Instant UI – zack zack
- Min. Klicks / Fachliche Aktion



# ➤ Unser Versprechen

## Dafür haben Sie bezahlt

### Skalierbarkeit

250 – 100 000  
Benutzer



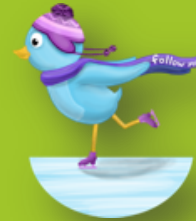
### Erweiterbarkeit

Architektur muss  
leicht erweitert werden  
können



### Leichtgewichtigkeit

Geringe  
Systemressourcen -  
Anforderungen



### Konfigurierbarkeit

Zentrale Änderungen  
der Laufzeit-  
konfigurationen



### Sicherheit

Verschlüsselte  
Kommunikation



### Clusterready

Die Anwendung soll  
im Cluster laufen  
können



### Monitoring

Überwachung der  
technischen und  
fachlichen Aktivitäten



### SLA

90% in 1s,  
100% in 3s

99,5 % Online



# ➤ Agenda:

1. Unser Versprechen
2. Unsere Lösung
3. Unsere Prozesse
4. Unsere Qualität
5. Unser Monitoring
6. Unsere Demo

# ➤ Unsere Lösung

Alles fängt mit einer Idee an

- 12 factor app  
(<http://12factor.net>)
- Containeransatz
- Backing Services
- Saubere  
Entwicklungsprozess
- Leichte Wartung



# ➤ Unsere Lösung

Alles fängt mit einer Idee an

- 12 factor app  
(<http://12factor.net>)
- Containeransatz
- Backing Services
- Saubere  
Entwicklungsprozess
- Leichte Wartung

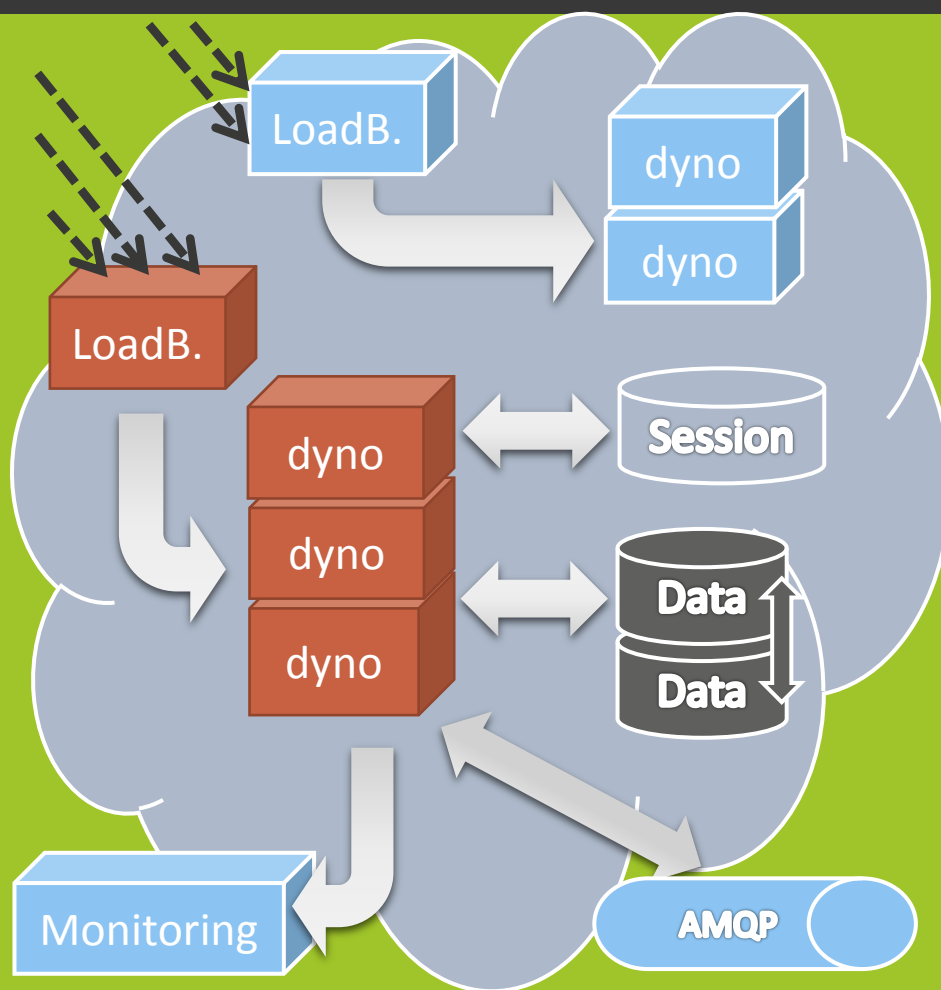


# ➤ Unsere Lösung

## Das haben Sie bekommen



### Leben im Cluster



### Highlights

- ✓ Leicht skalierbar
- ✓ Starke Einbeziehung von Cloud Diensten  
Sie zahlen nach einem festen Tarif
- ✓ Leichte Administration (Heroku Dashboard)
- ✓ Leicht erweiterbar (Wartungskosten)
- ✓ Ausgereiftes Monitoring (keine Mehrkosten)
- ✓ Extrem hohe Verfügbarkeit (Heroku & co.)

=  
**geringe Betriebskosten!**



# ➤ Unsere Lösung



Das haben Sie bekommen

## Rechenbeispiel

Leben im Cluster

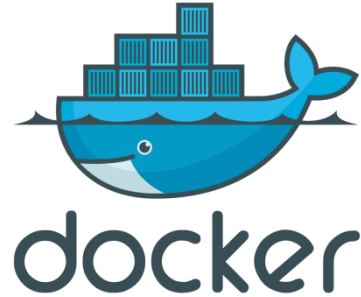
Parallele Benutzer / Dynozahl	100 2	1000 8	10000 20
Dyno-kosten	34 \$	250 \$	680\$
Monitoring	49\$	159\$	399\$
AMQP	19\$	99\$	299\$
Datenbanken	9\$+10\$	120\$+102\$	960\$+252\$
Support	Inkl.	Inkl.	Inkl.
<b>Monatlich ca.:</b>	<b>121 \$</b>	<b>703\$</b>	<b>2590\$</b>

Cloud-Diensten  
in Tarif  
(Heroku Dashboard)  
(Kosten)  
(ne Mehrkosten)  
(Heroku & co.)

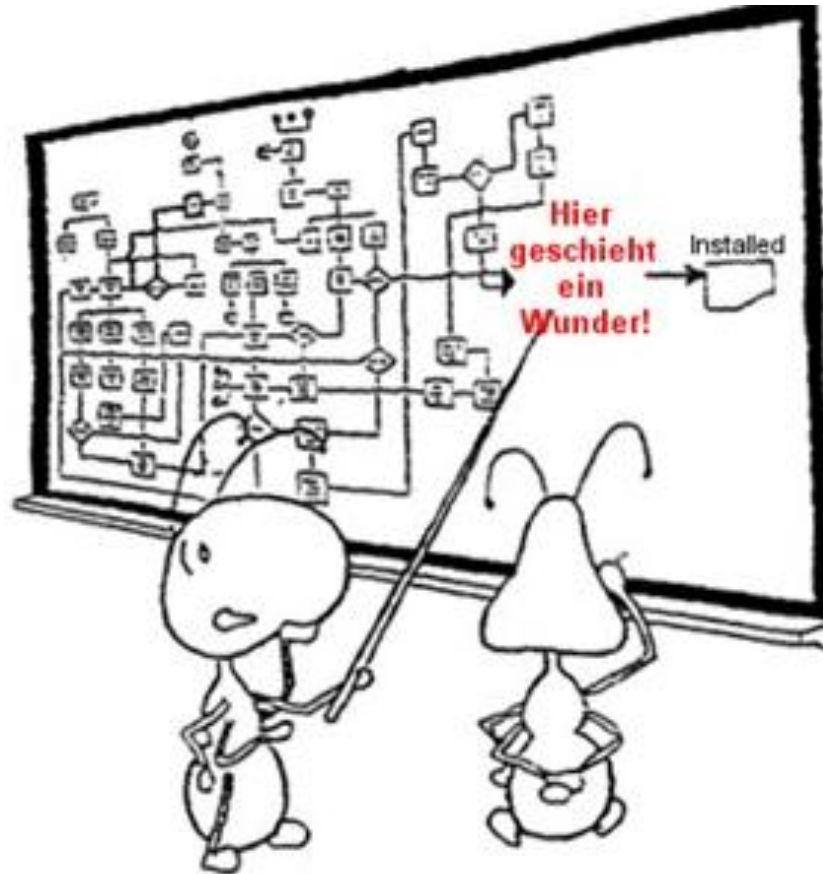
**geringe Betriebskosten!**

# ➤ Unsere Lösung

Das haben wir eingesetzt



# ➤ Unsere Lösung

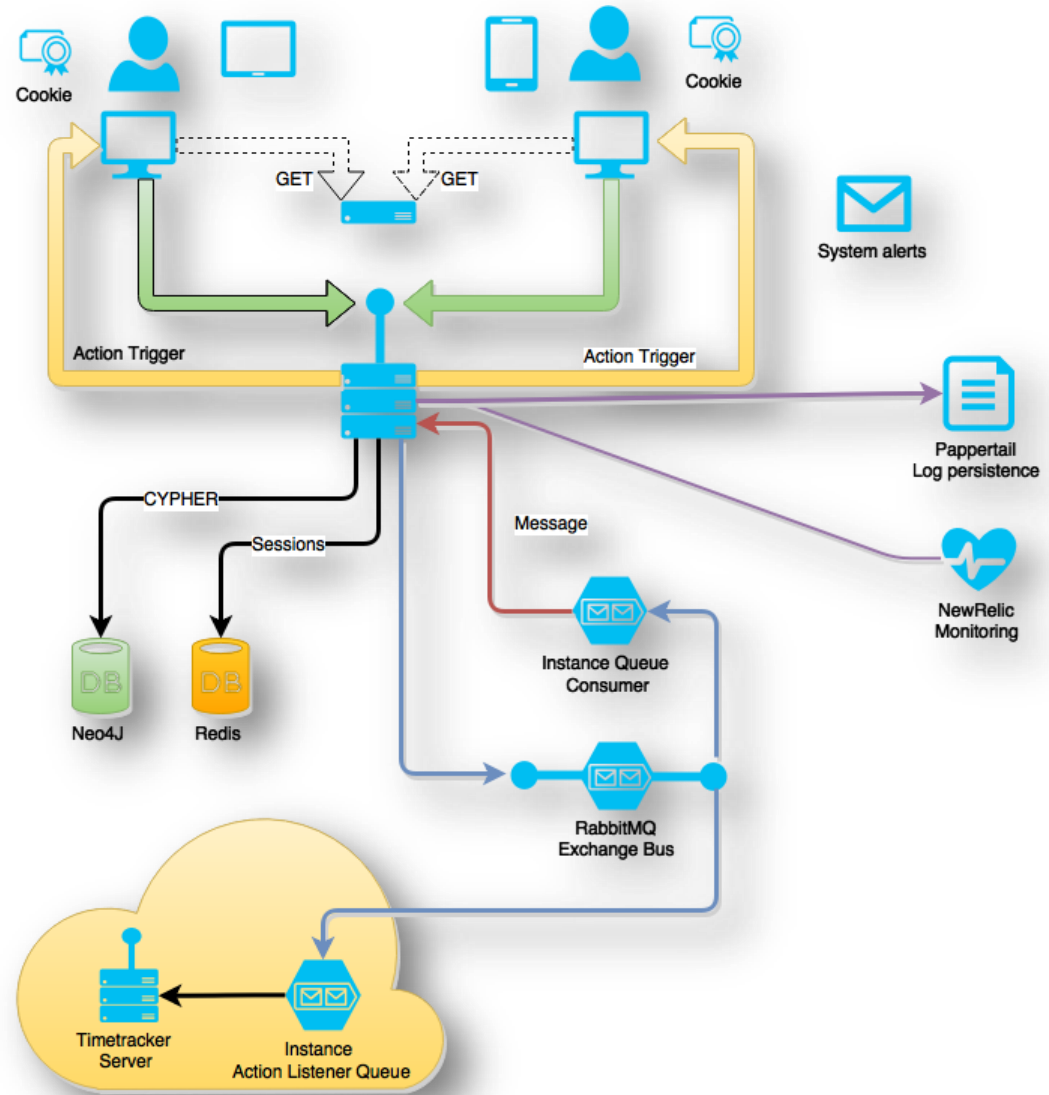


**Sehr gute Arbeit!  
Aber sollten wir hier nicht vielleicht  
etwas detaillierter werden...?**

# ➤ Unsere Lösung

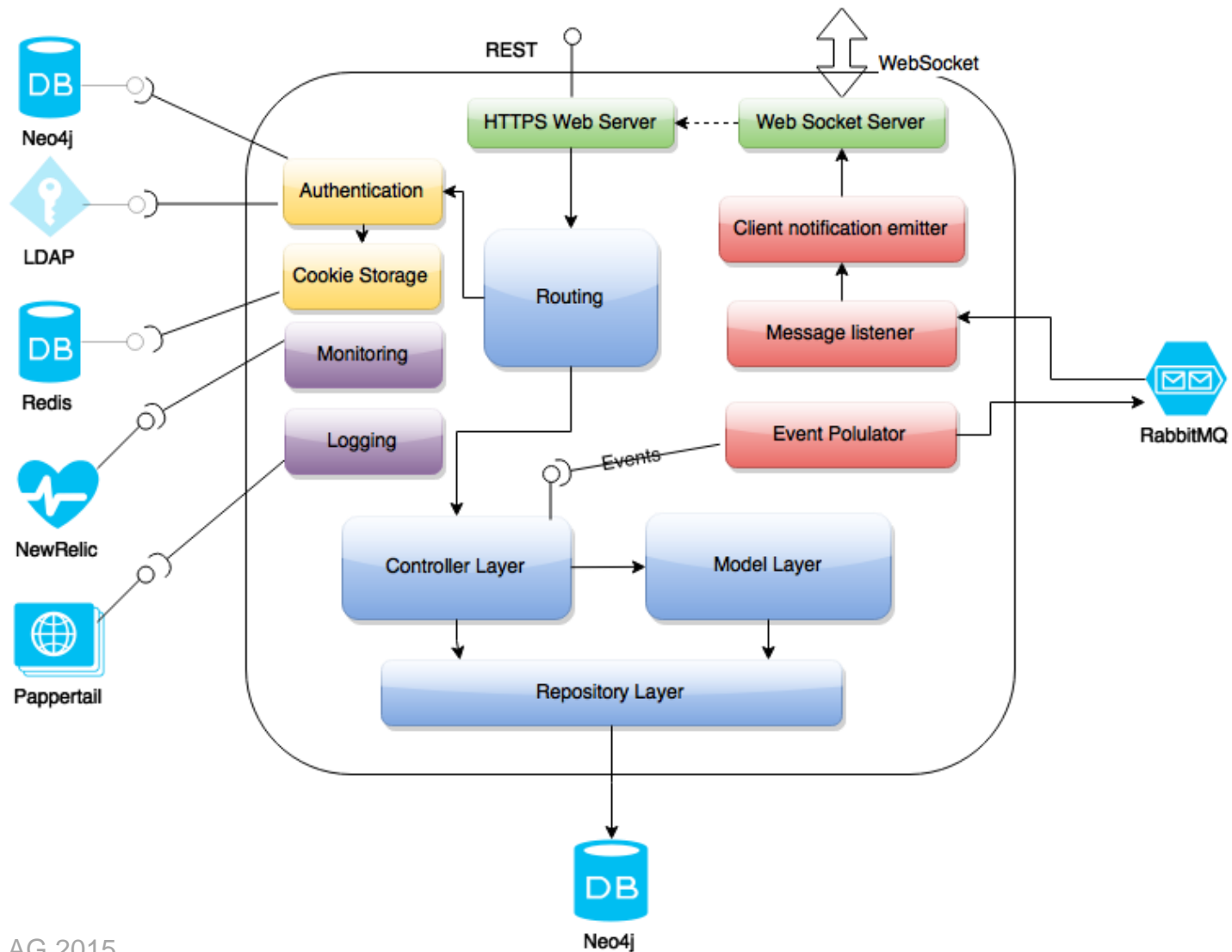
## Kommunikation wird bei uns Großgeschrieben

- Sitzungsdaten sind in Redis ausgelagert
- Benachrichtigung über Events mit AMQP (JMS) Messaging
- Aktive Monitoring von Fachaktivitäten
- Event Bus für interne Benachrichtigungen
- Websockets für Server Push



# ➤ Unsere Lösung

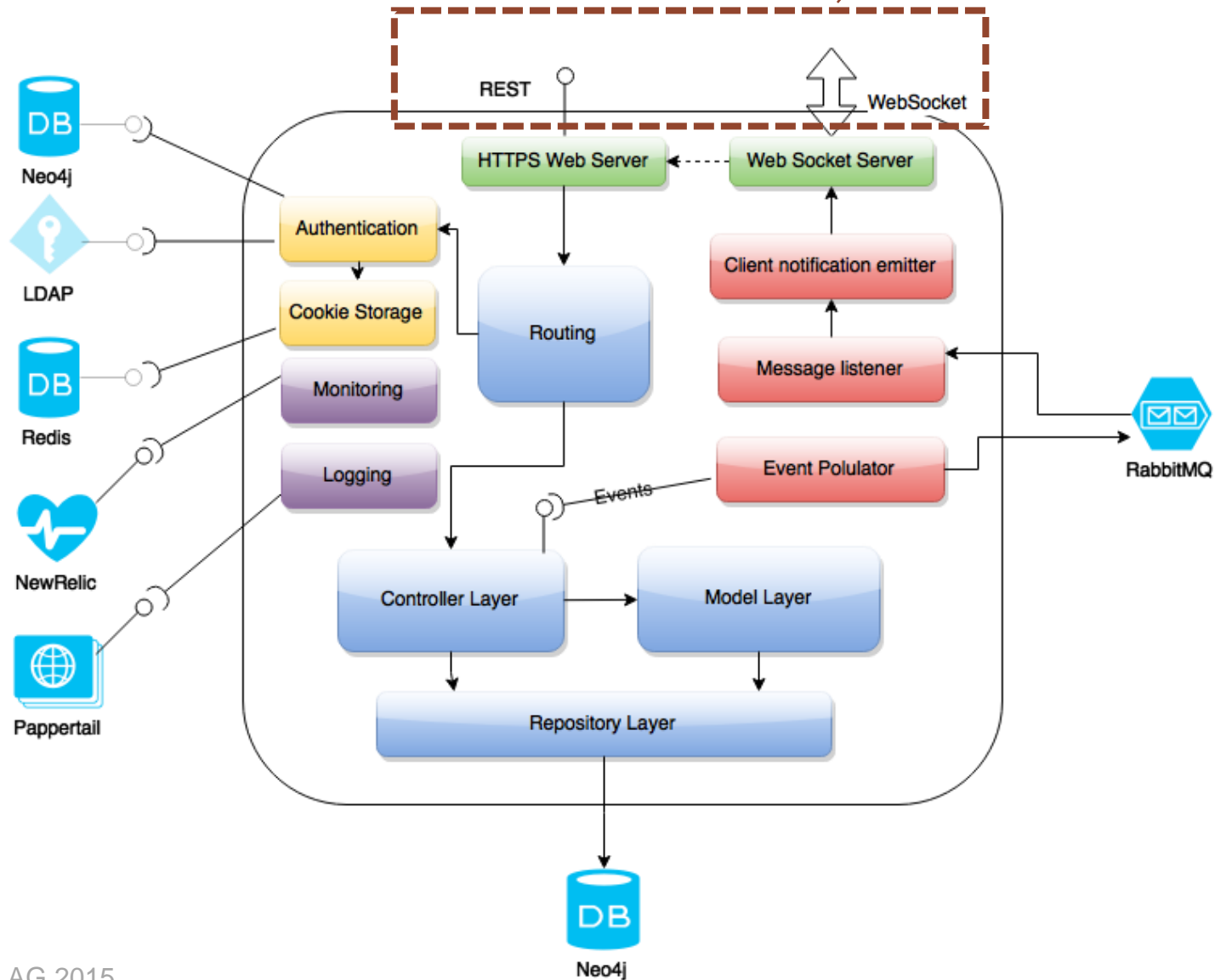
## Blick hinter die Kulissen



# ➤ Unsere Lösung

## Blick hinter die Kulissen

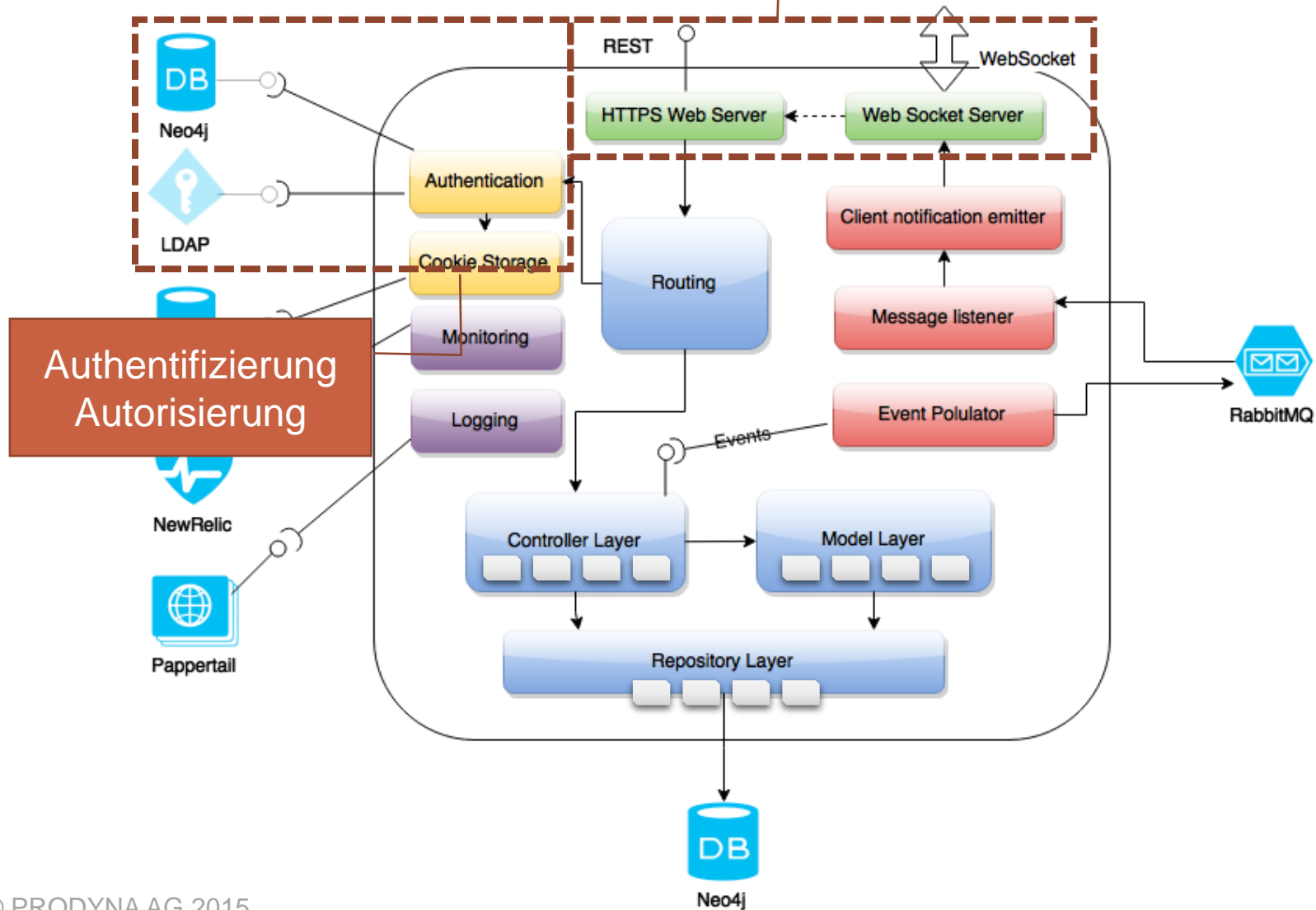
Lose Koppelung



# ➤ Unsere Lösung

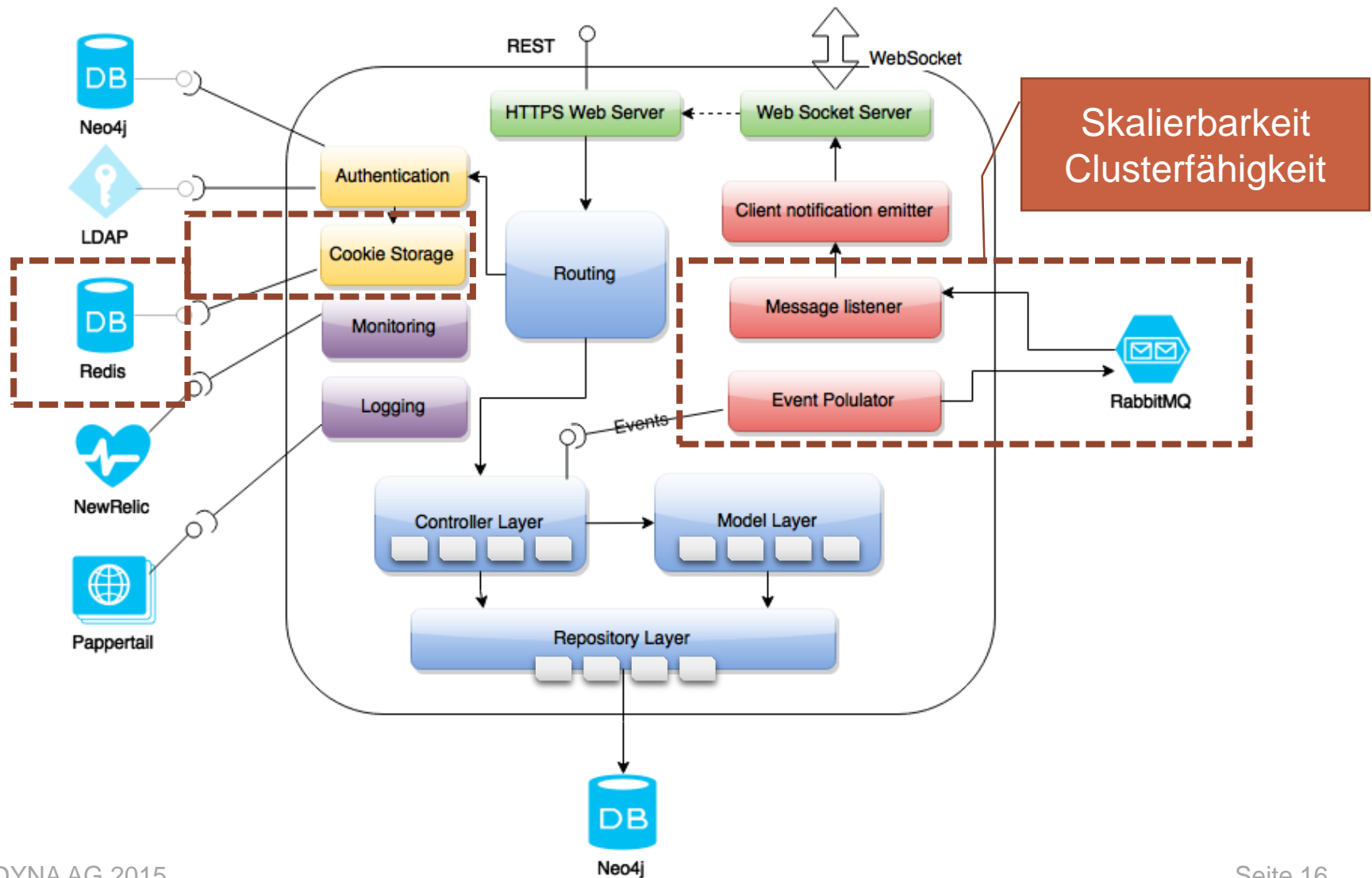
## Blick hinter die Kulissen

Verschlüsselte Kommunikation



# ➤ Unsere Lösung

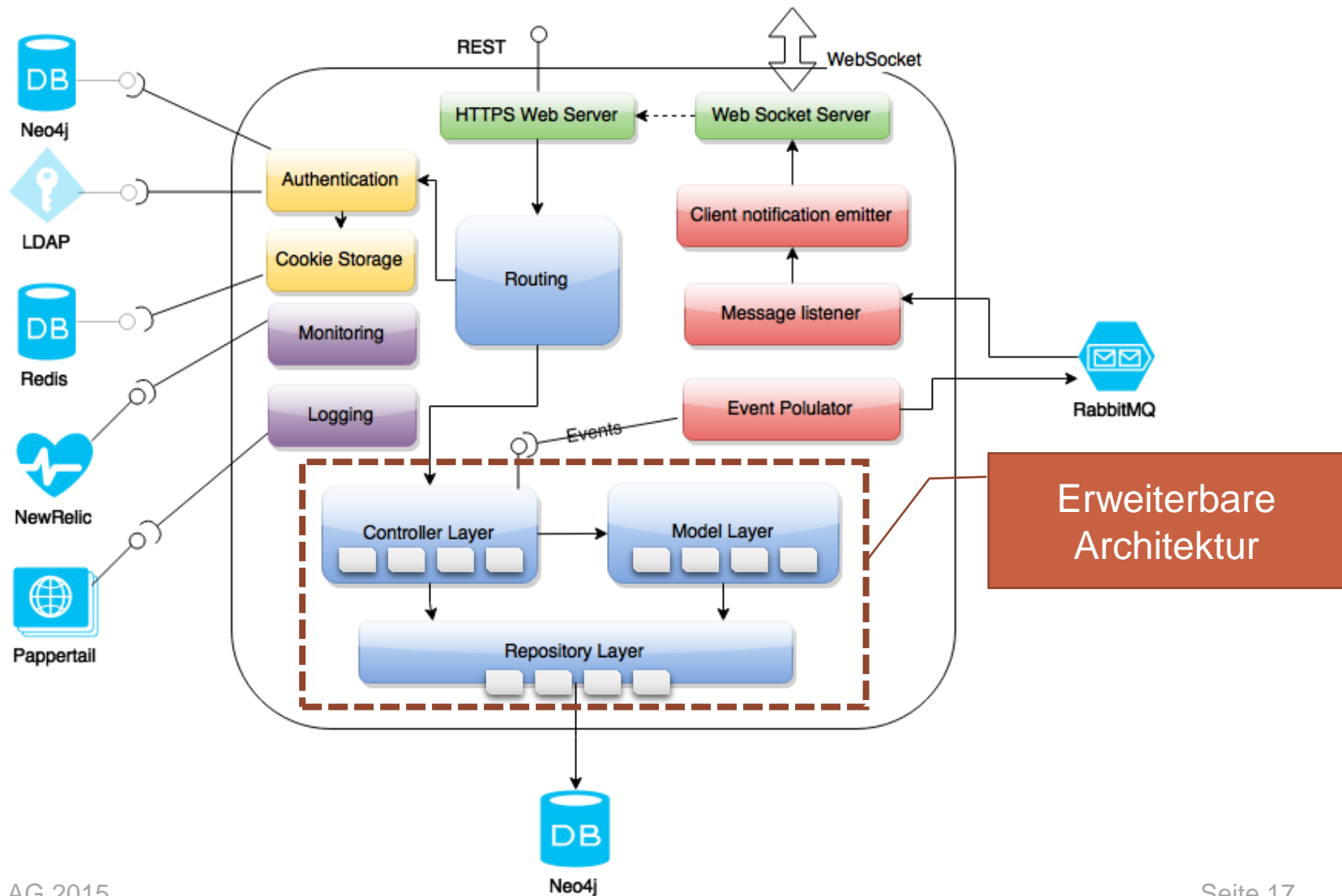
## Blick hinter die Kulissen





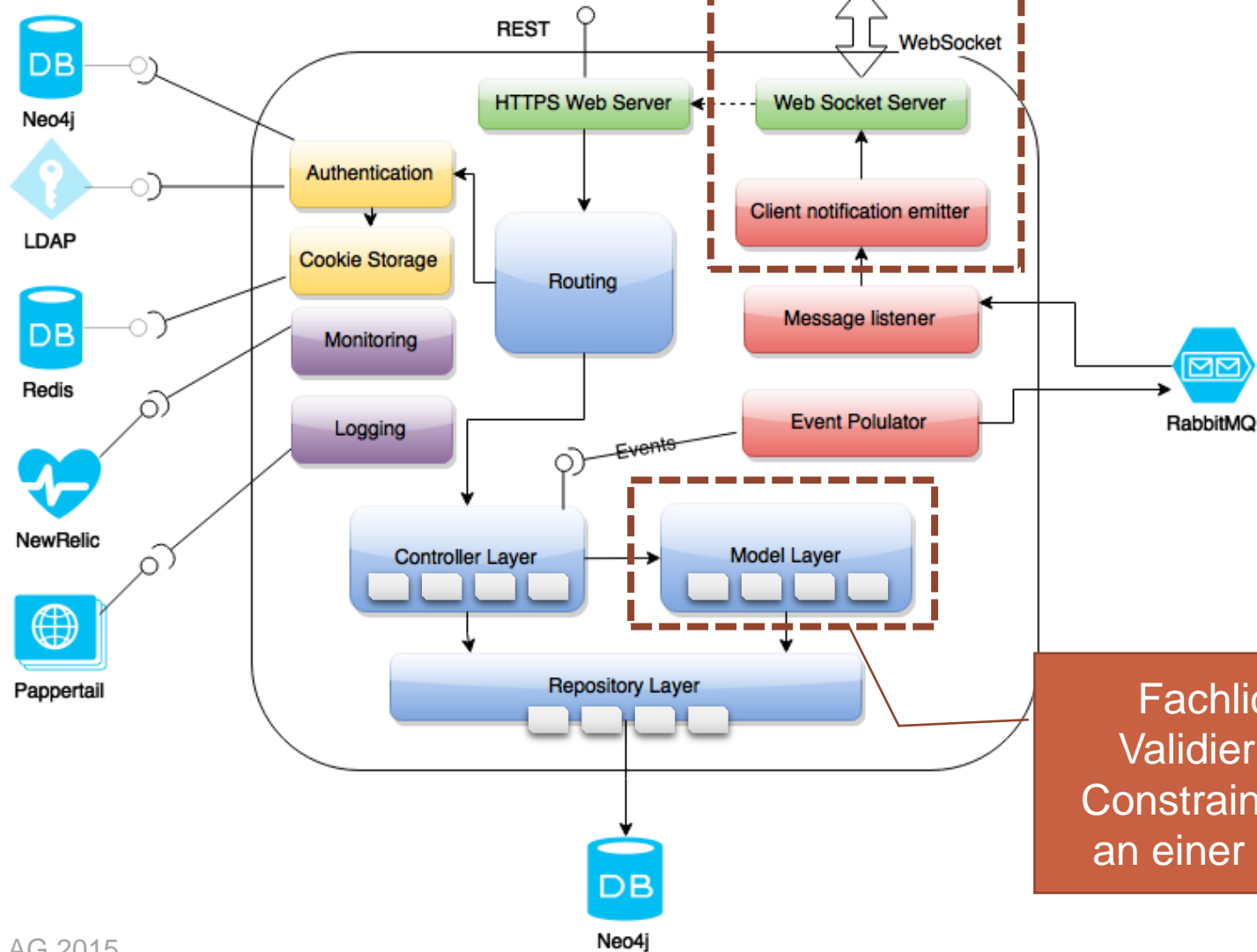
# ➤ Unsere Lösung

## Blick hinter die Kulissen



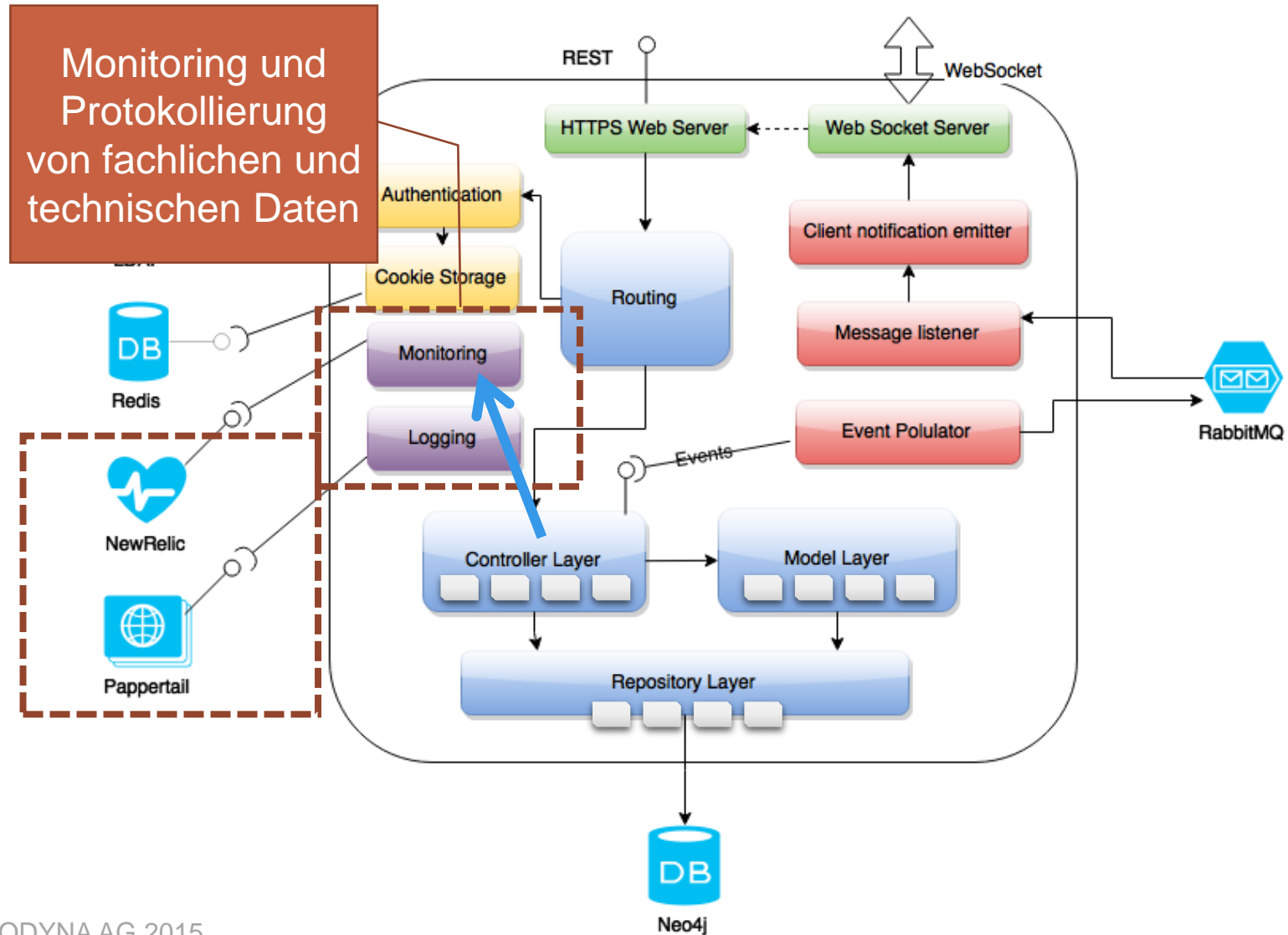
# ➤ Unsere Lösung

## Blick hinter die Kulissen



# ➤ Unsere Lösung

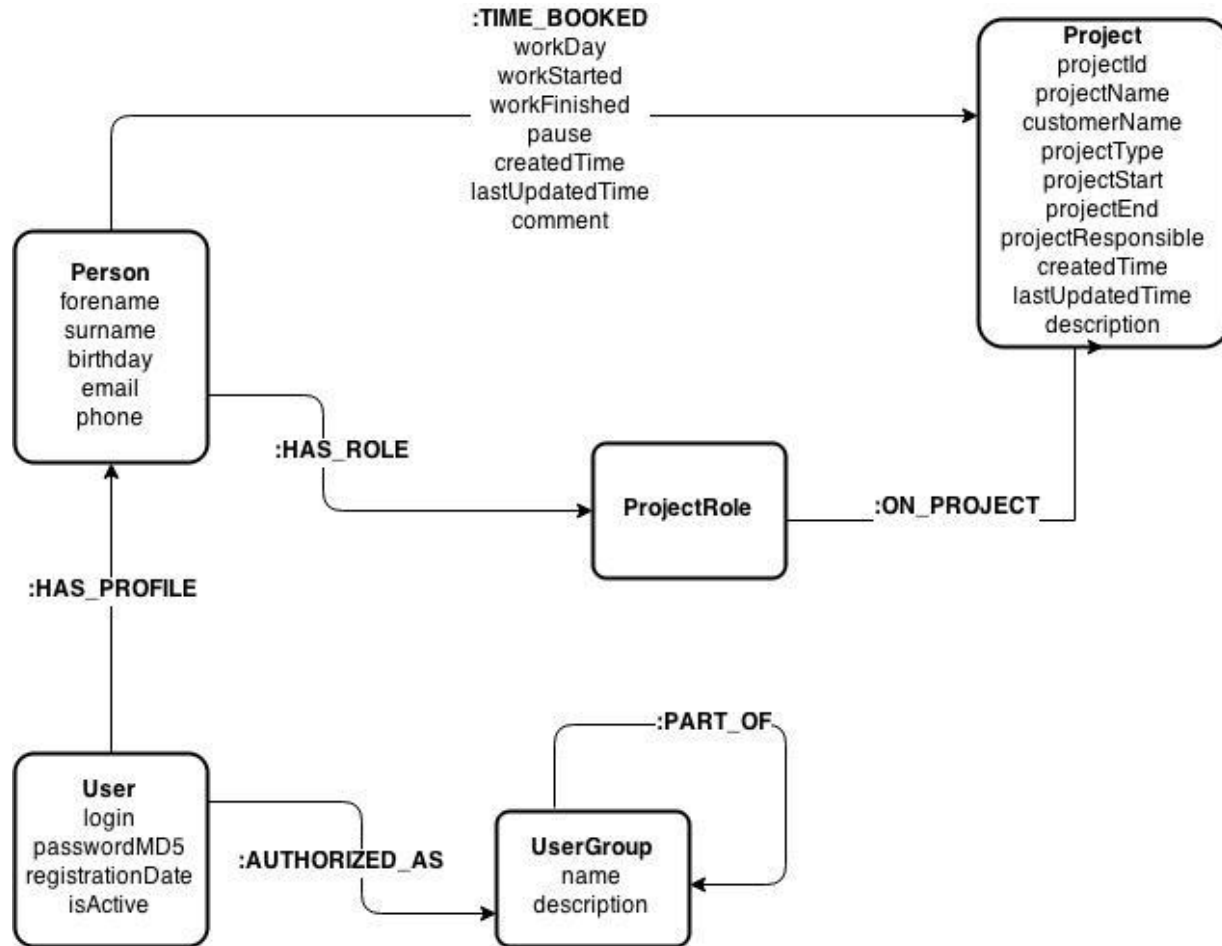
## Blick hinter die Kulissen



# ➤ Unsere Lösung

## Blick hinter die Kulissen

- Leicht erweiterbar
- Dynamische Authorisierung
- Trennung zwischen Fachlichkeit und Technik (Person und Benutzer)

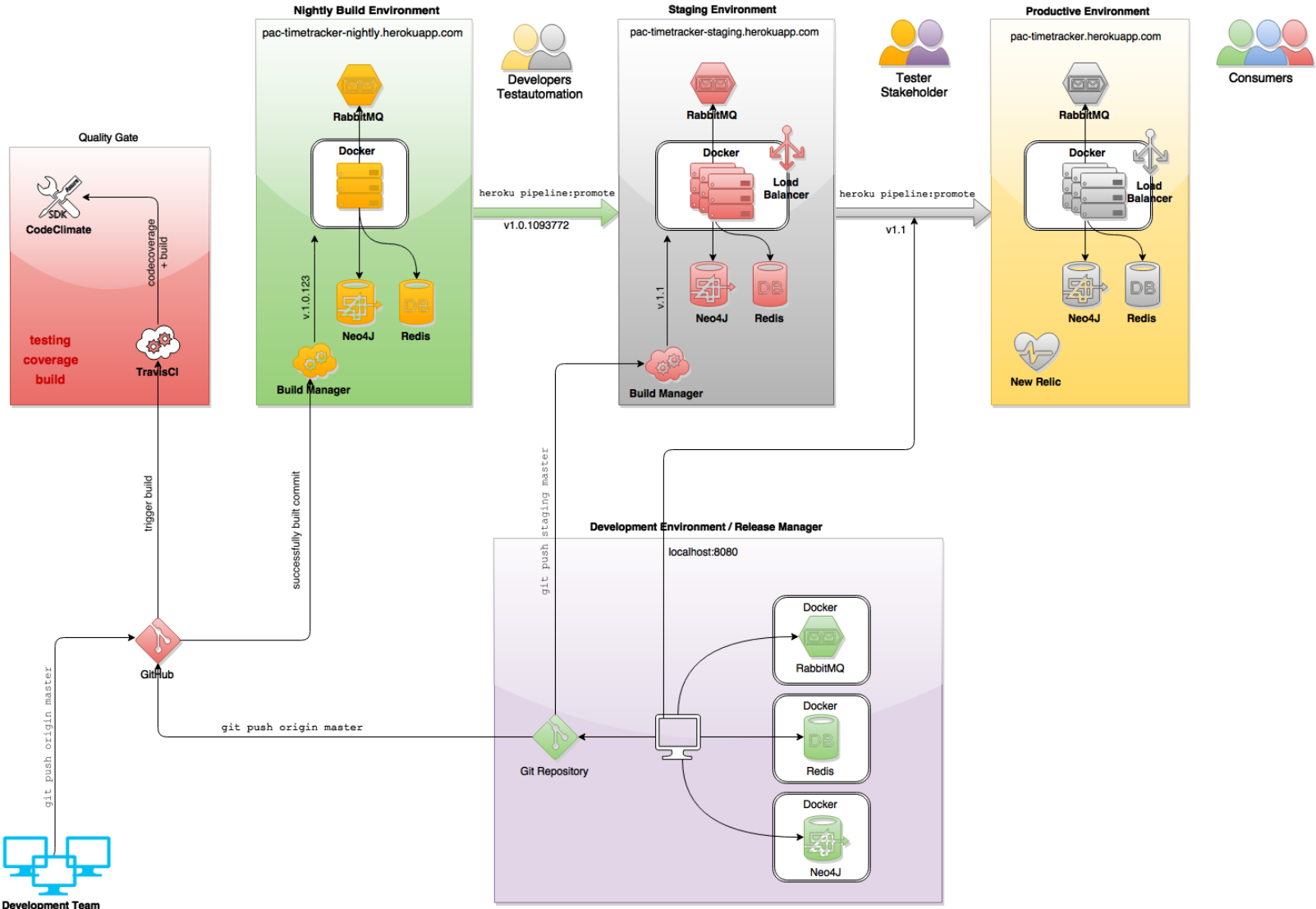


# ➤ Agenda:

1. Unser Versprechen
2. Unsere Lösung
3. Unsere Prozesse
4. Unsere Qualität
5. Unser Monitoring
6. Unsere Demo

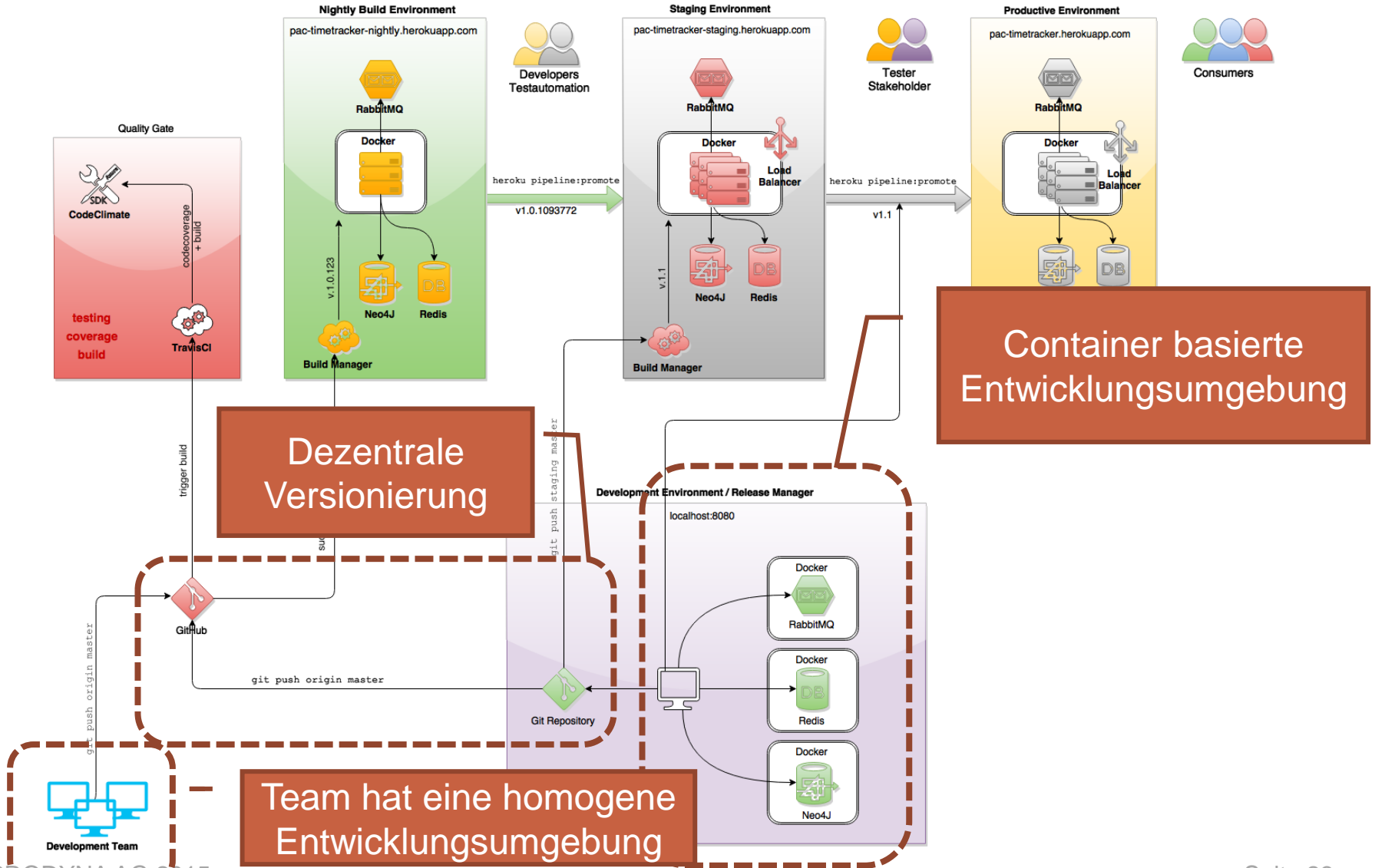
# ➤ Unsere Prozesse

## Klare Prozesse – Schlüssel zu Ihrem Erfolg



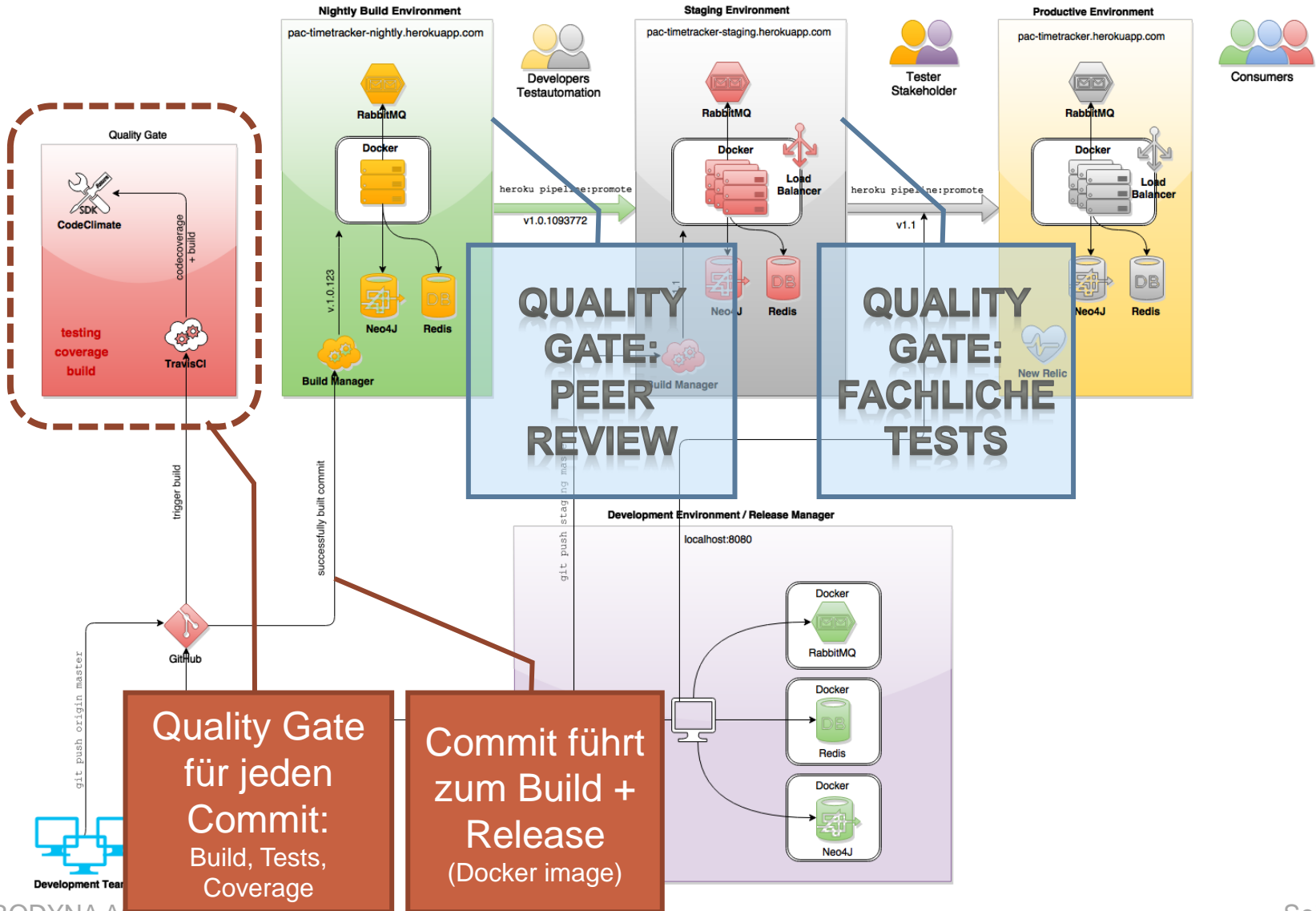
# ➤ Unsere Prozesse

## Klare Prozesse – Schlüssel zu Ihrem Erfolg



# ➤ Unsere Prozesse

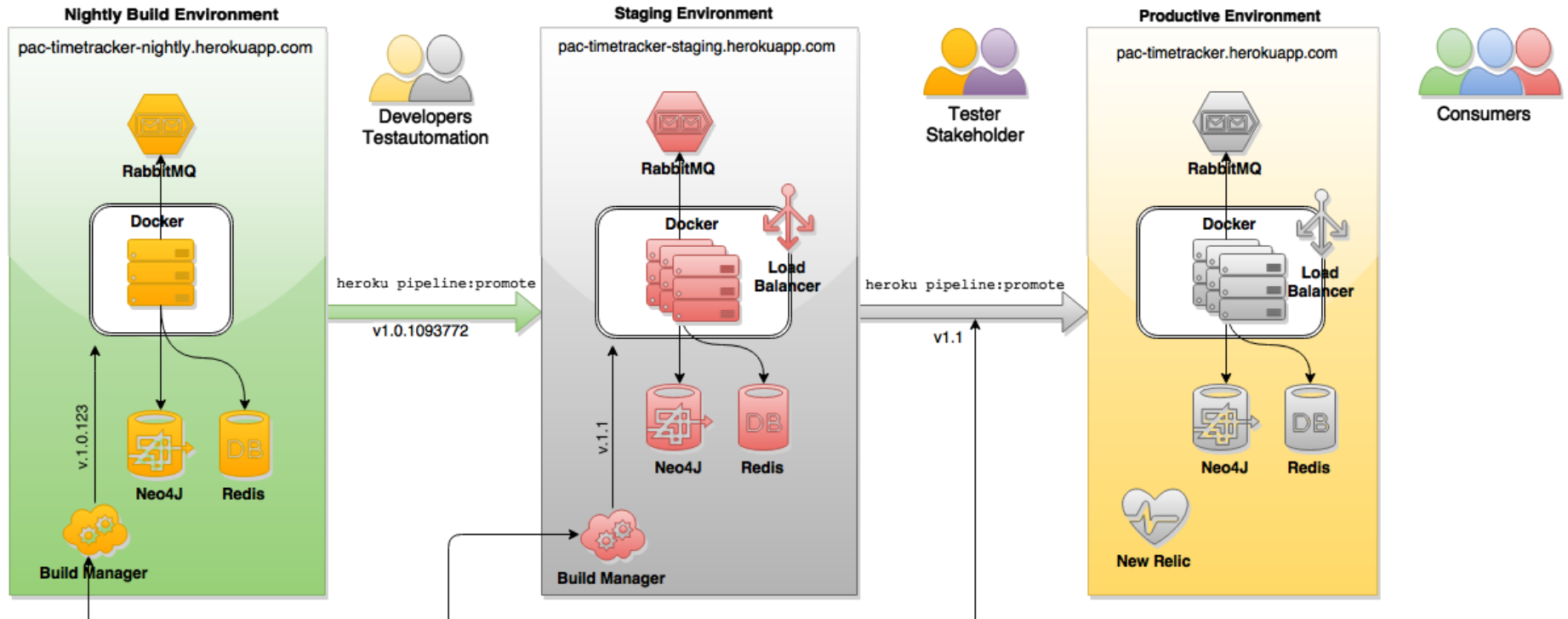
## Klare Prozesse – Schlüssel zu Ihrem Erfolg





# ➤ Unsere Prozesse

## Klare Prozesse – Schlüssel zu Ihrem Erfolg



- Docker basierte Release Pipeline `heroku pipeline:promote`
- Aufteilung in Nightly, Staging, Live Umgebungen
- Feedback vom Fachbereich im Staging area

# ➤ Unsere Prozesse

## Klare Prozesse – Schlüssel zu Ihrem Erfolg

### Nightly (Test)

- Technische Testumgebung
- Entwickler Commit führt über Quality Gate zum Release
- Schnelles Feedback
- Mehr Transparenz
- Frühere Fehlererkennung

### Staging

- Technische und fachliche Testumgebung
- Stelle für fachliche Testszenarien
- Abnahme und Freigabe für ->
- Mehrere Dyno's
- Identisch mit Nightly und Live

### Live

- Hochverfügbar
- Dauerüberwachung
- Alarmierung
- Hochskaliert
- Rollback falls Probleme mit neuen Version

# ➤ Agenda:

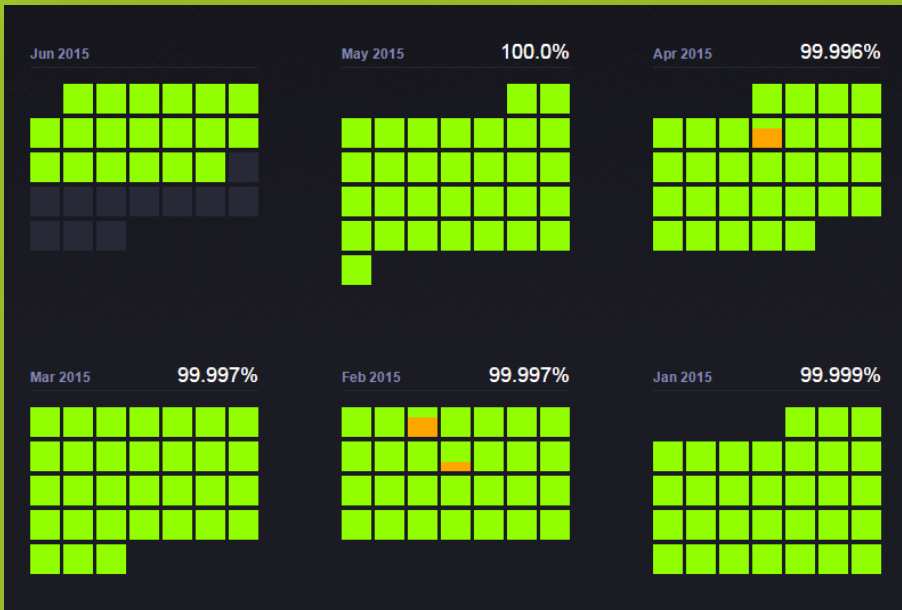
1. Unser Versprechen
2. Unsere Lösung
3. Unsere Prozesse
4. Unsere Qualität
5. Unser Monitoring
6. Unsere Demo

# ➤ Unsere Qualität

## Wir setzen auf beste Pferde



### Verfügbarkeit



Anwendung ist online solange alle seine Komponenten online sind

```
$ heroku ps:scale web=10 worker=1  
Scaling dynos... Done...
```



- PAAS
- Dynamisch skalierbar
- Integrierte Load Balancer
- Hochverfügbar 99,999%



- Monitoring
- Auswertung von Server und Client side Daten
- Auswertung von fachlichen Aktionen

### CloudAMQP

- Zuverlässige Messaging
- Lose Kopplung für Cluster
- Hochverfügbar 99,95%

### papertrail

- Logs Persistierung
- Suche & Analyse
- Alarm beim Auftreten von vordefinierten Muster

# ➤ Unsere Qualität

Wir setzen auf beste Pferde



## DB als Backing Service



Business Daten

Session Daten

## GRAPHENEDB

- Amazon hosted
- Dynamisch skalierbar
- Eigenes Monitoring
- Support + SLA

## redislabs

- Multi-core Redis
- Replikation
- Auto-failover
- Data persistence
- Backups
- Support

# ➤ Unsere Qualität

Vertrauen ist gut, Tests mit Testabdeckung sind besser

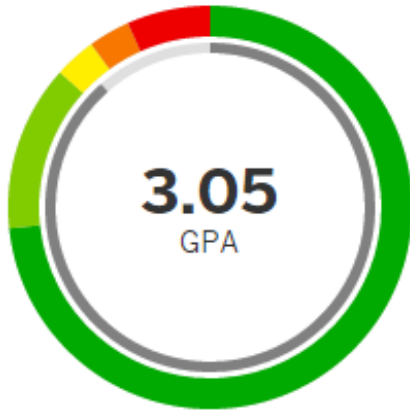
## 88.48% Test Coverage

Path	Coverage	Relevant LOC	Covered	Missed	Hits / Line
app/config/config.js	100.0%	4	4	0	1.0
app/controller/AssignmentController.js	80.0%	40	32	8	0.8
app/model/Booking.js	100.0%	18	18	0	25.6
app/model/BookingModel.js	81.67%	120	98	22	2.0
app/model/BookingsRepository.js	81.67%	120	98	22	0.9
app/model/Entity.js	100.0%	5	5	0	25.6
app/model/Person.js	100.0%	13	13	0	13.2
app/model/PersonRepository.js	87.23%	47	41	6	1.1
app/model/Project.js	100.0%	18	18	0	16.2
app/model/ProjectAssignmentRepository.js	97.37%	38	37	1	1.9
app/model/ProjectModel.js	88.24%	85	75	10	1.6
app/model/ProjectRepository.js	98.41%	63	62	1	1.5
app/model/User.js	100.0%	16	16	0	26.4
app/model/UserModel.js	90.71%	140	127	13	2.3
app/model/UserRepository.js	87.83%	115	101	14	2.2

- Unit Tests mit Mocha+Sinon (Mocks)
- Integrationstests mit Mocha + Supertest (Http)

# ➤ Unsere Qualität

Vertrauen ist gut, statische Codeanalyse ist besser



code climate 3.1 coverage 88%  
Link to Code Climate from your README.

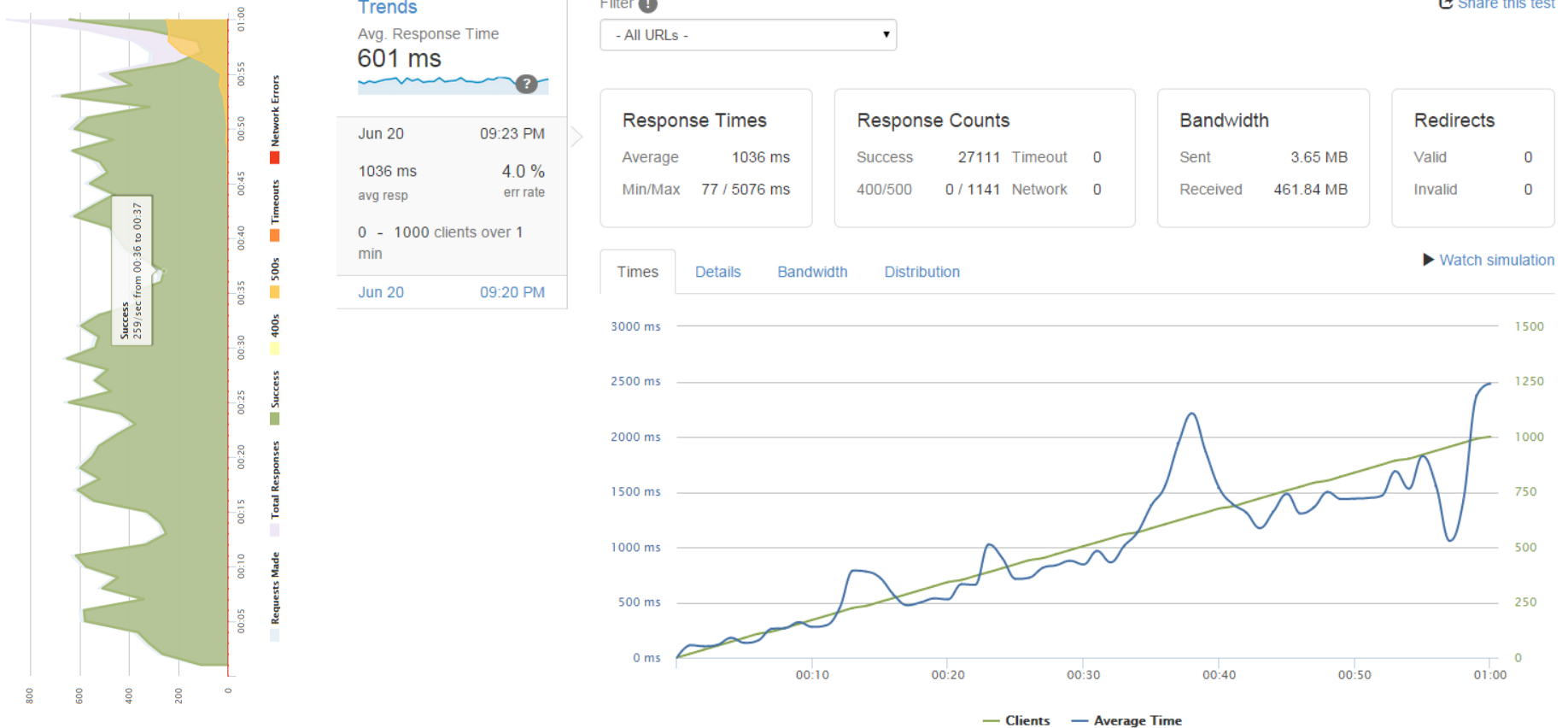
 88.5% Test Coverage

Rating	Name	LOC	Duplication	Churn	Issues
B	app/app.js	81	0	36	1
A	app/config/config.js	26	0	14	0
A	app/config/eventPopulator.js	63	0	5	0
A	app/config/passport.js	19	0	3	0
B	app/config/routes.js	78	0	20	1
A	app/config/sockets.js	68	0	9	0
A	app/controller/AssignmentController.js	58	0	7	0
C	app/controller/BookingController.js	117	92	11	2
A	app/controller/DataController.js	34	0	4	0
B	app/controller/PersonController.js	46	66	7	2

- Analyse von Codequalität (Style, Complexity etc.)
- Jedes Commit -> Code Analyse
- Erzeugung von Qualitätsreports

# ➤ Unsere Qualität

## Vertrauen ist gut, Lasttests sind besser



Performance Probleme werden frühzeitig entdeckt



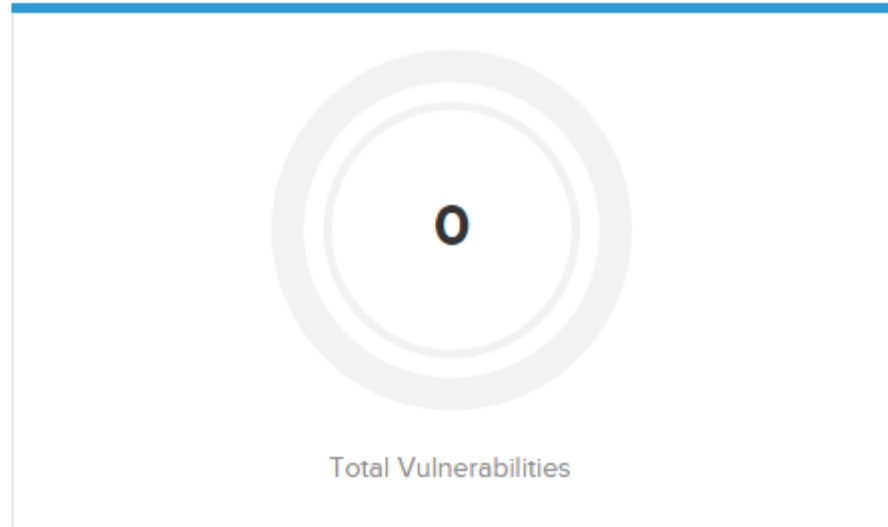
# ➤ Unsere Qualität

Vertrauen ist gut, Sicherheitstests sind besser

STATUS ON 06/11/2015



NUMBER OF VULNERABILITIES



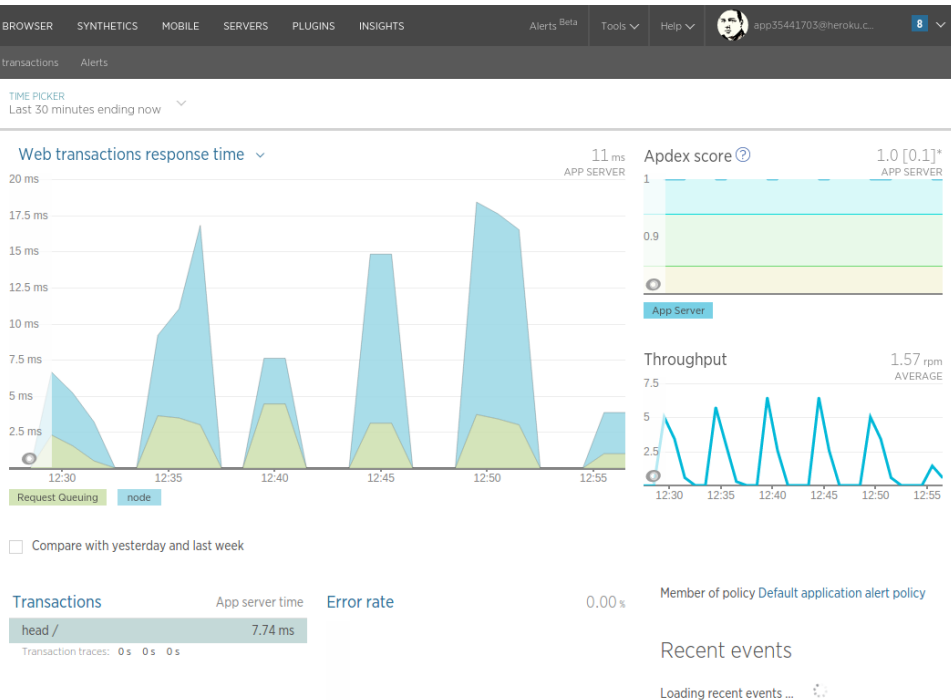
- Regelmäßige Sicherheitsüberprüfung
- Suche nach bekannten Sicherheitslücken
- Erzeugung von Sicherheitsreports

# ➤ Agenda:

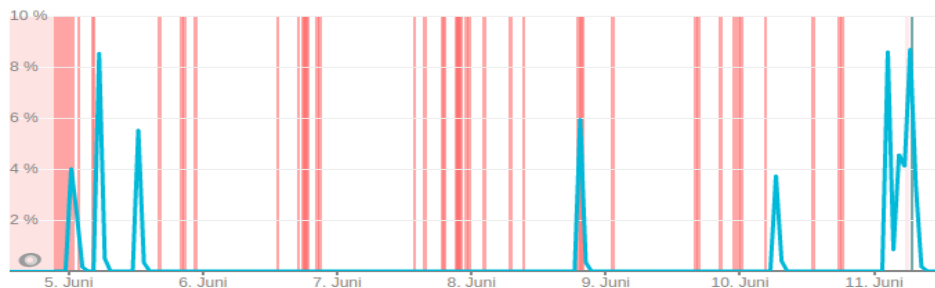
1. Unser Versprechen
2. Unsere Lösung
3. Unsere Prozesse
4. Unsere Qualität
5. Unser Monitoring
6. Unsere Demo

# ➤ Unser Monitoring

## Zahlen sagen mehr als Worte, wenn man sie versteht...



### Error rate (errors per request)



### New Relic

- Auswertung von: Antwortzeiten, Durchsatz, Fehlerquote usw...
- Speicherung und Visualisierung
- Active Ping
- Visualisierung von Daten aus:
  - Server
  - Fachliche Events
  - Browserverhalten
  - Hardware

# ➤ Unser Monitoring

Zahlen sagen mehr als Worte, wenn man sie versteht...

## Application Performance Index

- Objektive Wert zum messen der Nutzerzufriedenheit
- Kann benutzt werden um Rückschlüsse auf Skalierbarkeit zu ziehen

$$\text{APDEX} = \frac{\text{SATISFIED} + \frac{\text{TOLERATED}}{2}}{\text{TOTAL USERS}}$$

$$(100 + 50 / 2) / 150 = 0,83 > \text{ok}$$

$$(50 + 100 / 2) / 150 = 0,66 > \text{warn}$$

$$(50 + 100 / 2) / 200 = 0,5 > \text{danger}$$

## Application Policy

### Default application alert policy (default)

#### 2 applications

New applications are placed here by default.

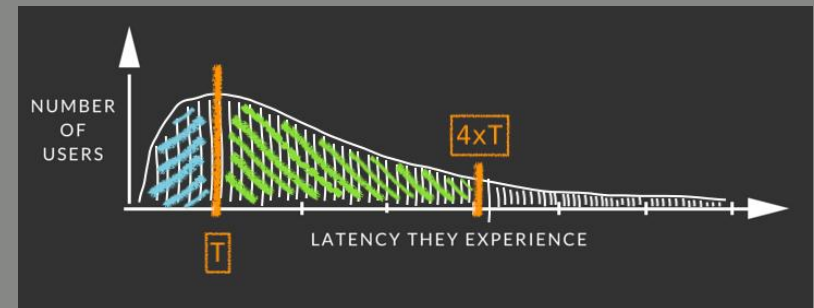
- timetracker-server
- timetracker-client

#### Alerts when:

APDEX  $\leq 0.7$       ERROR RATE  $> 5.0\%$

Downtime alerts **ON**

▲ 1 application is missing a ping



# ➤ Unser Monitoring

Zahlen sagen mehr als Worte, wenn man sie versteht...

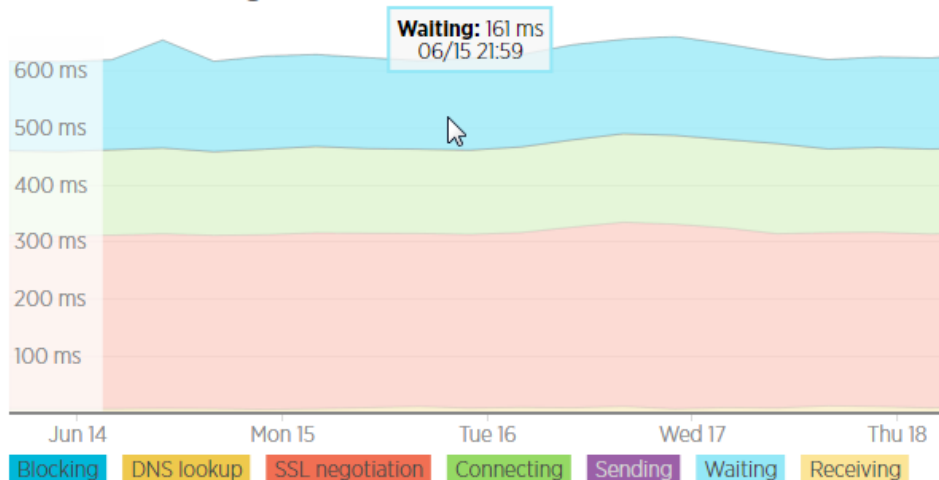
## Network Time

- Auswertung von Verbindungszeiten
- Reaktionszeiten nachvollziehen

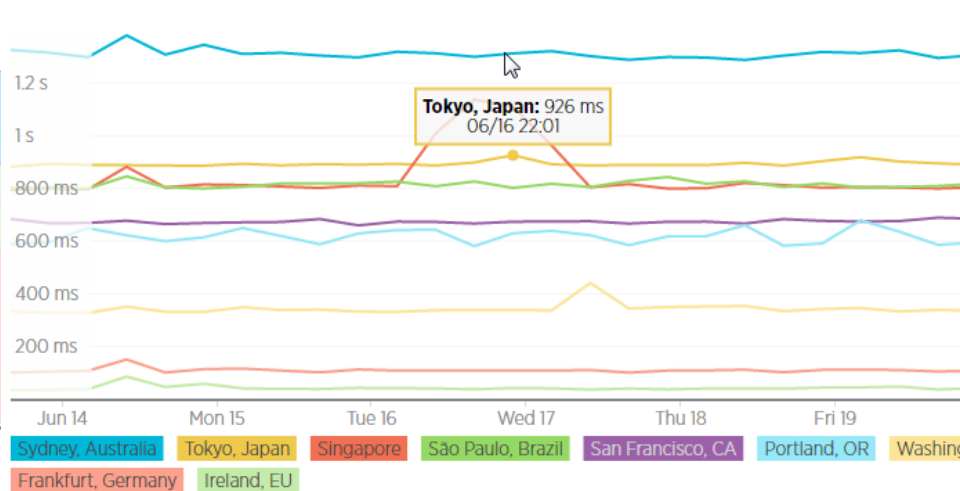
## Load time

- Weltweites Zeitbasierte Ping
- Auswertung der Antwortzeiten für bestimmte Regionen

### Network timings



### Load time

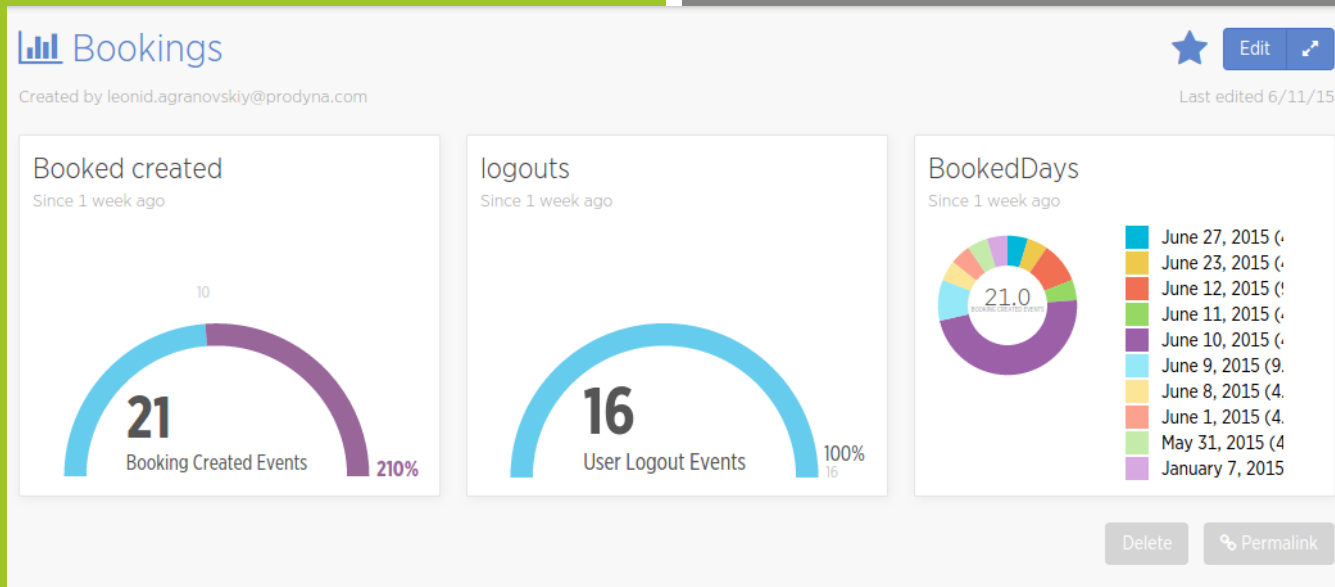


# ➤ Unser Monitoring

Zahlen sagen mehr als Worte, wenn man sie versteht...

- Auswertung von fachlichen Events (z.B. Buchung erzeugt)
- Events werden aktiv aus den Anwendung gesendet

- Dashboards mit NSQL
- Überwachung und Persistierung
- Alerts and policies



# ➤ Unser Monitoring

## Worte sagen auch einiges aus

### Dashboard

**11,000 events**

last seen moments ago

#### Searches

All events

Deploys

Dyno state changes

Error

Platform errors

Scheduler jobs

Web app output

### Latest Archives

2015-06-24.tsv.gz (175.9 KB)

Download

2015-06-23.tsv.gz (180.9 KB)

Download

2015-06-22.tsv.gz (175.6 KB)

Download

[All Archives »](#)

### Log Data Transfer



**8% used**

Estimated total: 952.4 KB

- Log Analyse
- Filterung
- Persistierung
- Alerts bei Events

pac-timetracker

Dashboard

Alerts

Events

Account

Help

```
Jun 25 14:35:20 pac-timetracker heroku/router: at=info method=HEAD path="/" host=pac-timetracker.herokuapp
request_id=4c3bee6a-9fbf-416c-9ac4-2dbd16b125fb fwd="54.78.163.26" dyno=web.1 connect=0ms service=3ms
status=200 bytes=541
Jun 25 14:35:20 pac-timetracker heroku/router: at=info method=HEAD path="/" host=pac-timetracker.herokuapp
request_id=b61307be-0d87-480f-8e0e-5b10f3a4ff91 fwd="54.232.5.179" dyno=web.1 connect=0ms service=3ms
status=200 bytes=547
Jun 25 14:35:20 pac-timetracker app/web.1: HEAD / 200 57 - 2.262 ms
Jun 25 14:40:10 pac-timetracker heroku/router: at=info method=HEAD path="/" host=pac-timetracker.herokuapp
request_id=c8fd5e3c-0ea0-4db2-95f3-140920d8c35e fwd="52.28.150.205" dyno=web.1 connect=0ms service=4ms
status=200 bytes=547
Jun 25 14:40:11 pac-timetracker app/web.1: HEAD / 200 57 - 2.474 ms
Jun 25 14:40:11 pac-timetracker heroku/router: at=info method=HEAD path="/" host=pac-timetracker.herokuapp
request_id=f9054178-b5c4-4fca-af76-b8560e7b45bf fwd="54.206.171.182" dyno=web.1 connect=5ms service=3ms
status=200 bytes=543
Jun 25 14:40:11 pac-timetracker app/web.1: HEAD / 200 57 - 2.220 ms
Jun 25 14:40:11 pac-timetracker app/web.1: HEAD / 200 57 - 2.602 ms
Jun 25 14:40:12 pac-timetracker app/web.1: HEAD / 200 57 - 2.375 ms
Jun 25 14:40:13 pac-timetracker heroku/router: at=info method=HEAD path="/" host=pac-timetracker.herokuapp
request_id=353c47d9-fe0f-41f1-9e6a-487b769fc6c8 fwd="54.250.14.225" dyno=web.1 connect=0ms service=3ms
status=200 bytes=541
```



Example: H12 OR status=5 OR "Starting process" -png

Search



# ➤ Agenda:

1. Unser Versprechen
2. Unsere Lösung
3. Unsere Prozesse
4. Unsere Qualität
5. Unser Monitoring
6. Unsere Demo



# Demo



Show time

# Vielen Dank



Leonid Agranovskiy  
PRODYNA AG

IT Consultant / Software Engineer  
Tel. 0176 / 178 70 248

## ➤ 12 Factor app

- One codebase tracked in revision control, many deploys
- Explicitly declare and isolate dependencies
- Store config in the environment
- Treat backing services as attached resources
- Strictly separate build and run stages
- Execute the app as one or more stateless processes
- Export services via port binding
- Scale out via the process model
- Maximize robustness with fast startup and graceful shutdown
- Keep development, staging, and production as similar as possible
- Treat logs as event streams
- Run admin/management tasks as one-off processes

# ➤ Gezogene Lehre

Mut etwas neues Auszuprobieren

- Sehr breites Spektrum an neu gelernten Technologien
- Weitblick im Bezug auf wichtige Architekturthemen
- The Twelve Factor App  
<http://12factor.net/>
- Git/Docker basierte Releasemanagement
- Viel Erfahrung mit Unit Tests im JS Umfeld
- CodeClimate? Nein Danke
- Arbeit mit Datumswerten birgt mehr Probleme als man denkt
- DB Datenstrukturen könnten besser ausgenutzt werden