DNS Belgium in the cloud

ICANN Tech Day – 2017-03-13

maarten.bosteels@dnsbelgium.be

What did we do ?

- Migrated to Amazon Web Services

- Re-built entire registration platform from code

- Took down the wall between Ops & Dev

Main drivers for change

- Configuration drift (Test vs. Prod)

- Long lead times (eg. patching)

- Difficult hand-overs Dev-Ops

- Infrequent deployments

- Lots of fire fighting, little time for fire prevention

- Aging hardware

Classic model ?
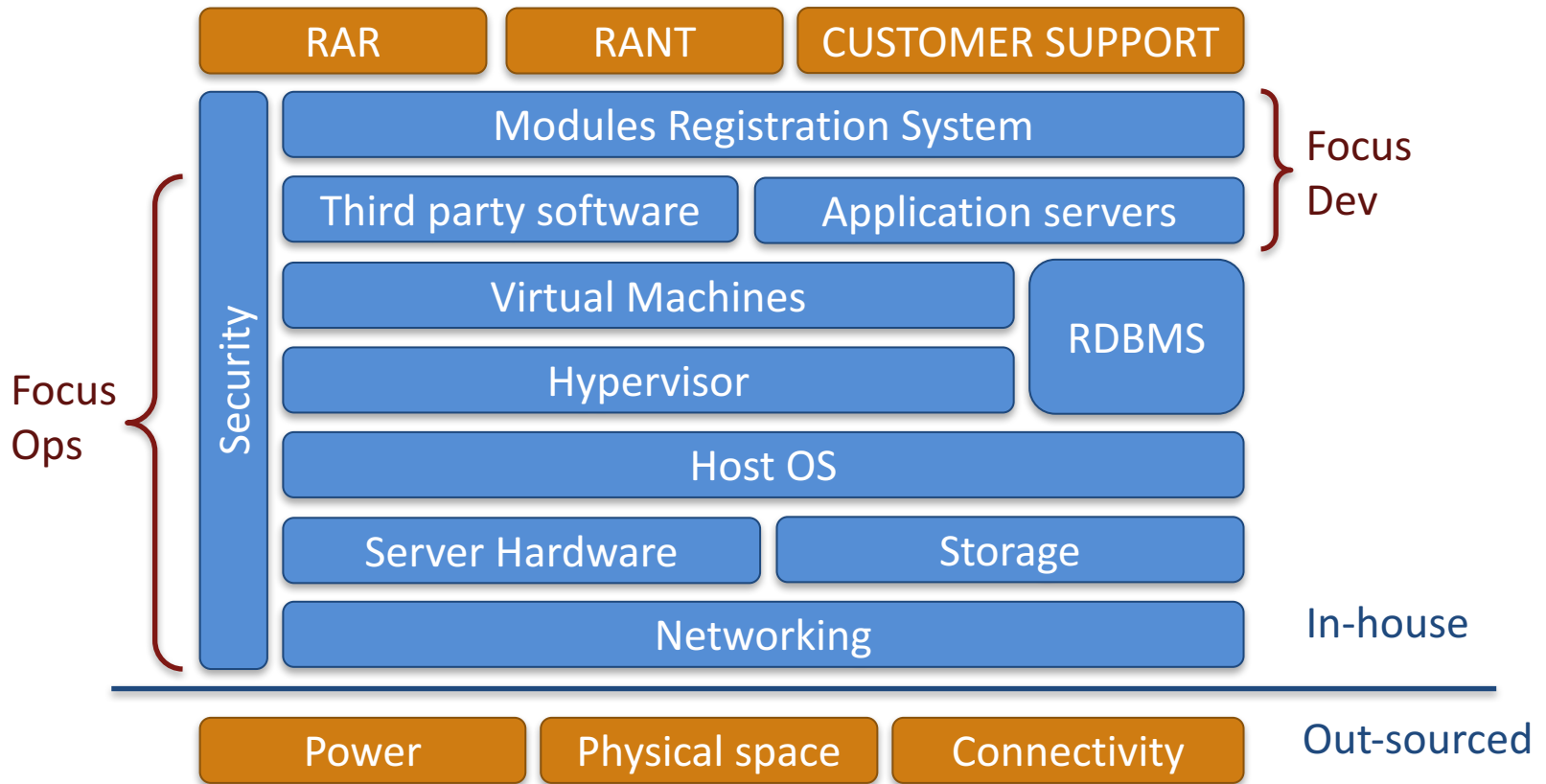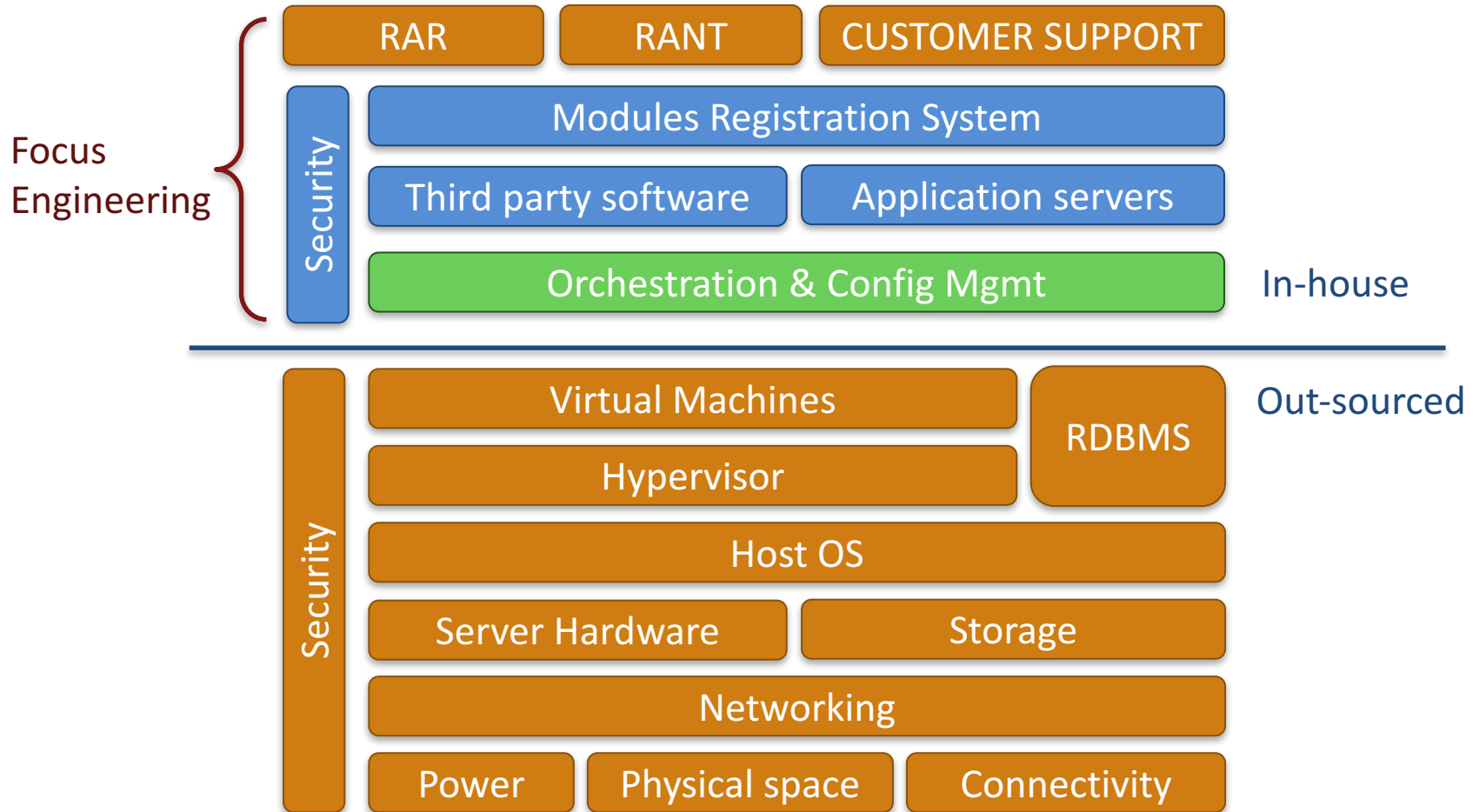
Engineering = Dev + Ops + QA

Strategy

- Multi-functional agile teams

- Focus on upper layers of the stack

- Infrastructure-as-code => **reproducible** & testable

- Continuous Delivery
  => small amounts of change & early feedback.

- Dev & Ops both confronted with quality of their work

- Design for failure: resilient, self-monitoring, self-healing

Status early 2015

- Last hardware renewal : 2011
  - Big bang migration
  - New hardware / network design / storage solution / colo
  - Lots of vendors to manage
- Go for another big bang ?
- Do we really need our own hardware ?
- Why not use the cloud ?

Extra layer

dnsbelgium

Focus
Engineering {

| Security | RAR | RANT | CUSTOMER SUPPORT |

| Modules Registration System |
| Third party software | Application servers |
| Orchestration & Config Mgmt | In-house

| Security | Virtual Machines | RDBMS | Out-sourced
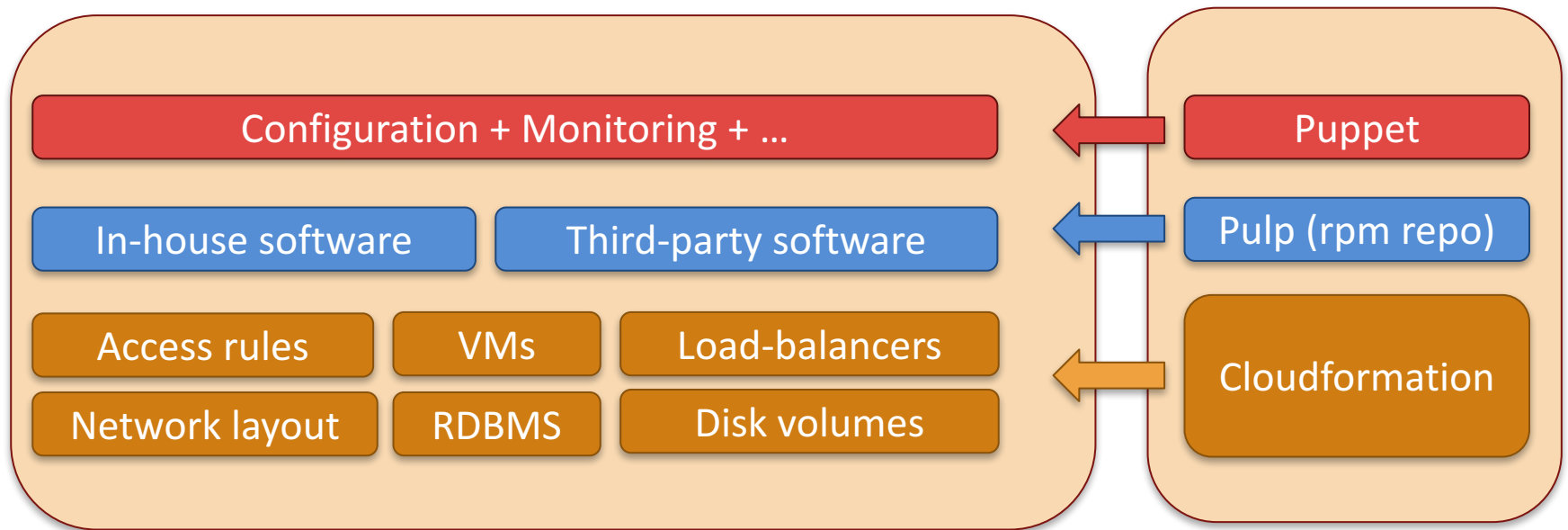| Hypervisor |
| Host OS |
| Server Hardware | Storage |
| Networking |
| Power | Physical space | Connectivity |

Initial assessment of AWS

- Initial tests:
    - Get to know AWS services
    - Proof-of-concept
- Risk assessment
    - Technically feature complete ?
    - Confidentiality, Integrity, Availability ?
    - Legal risk assessment
- Performance tests
- Cost assessment
    - Man days
    - €

Conclusion of assessment

- Software-defined everything
    - Avoid configuration drift
    - Infra predictable & documented => increased security

- Encryption all data in transit + data at rest

- IaaS = enabler to focus on core business
    - No need for home-grown HA solutions

    - Use well-designed services with built-in redundancy

    - Underlying services keep improving 'for free'

- Pay what you use
   - Dev & Test environments : business hours only

    - Easily scale up / down

Infra-as-code: building blocks

| Configuration + Monitoring + … | ← | Puppet |

| In-house software | Third-party software | ← | Pulp (rpm repo) |

| Access rules | VMs | Load-balancers | |
| Network layout | RDBMS | Disk volumes | ← | Cloudformation |

Git repos:
- Puppet modules + config
- In-house software
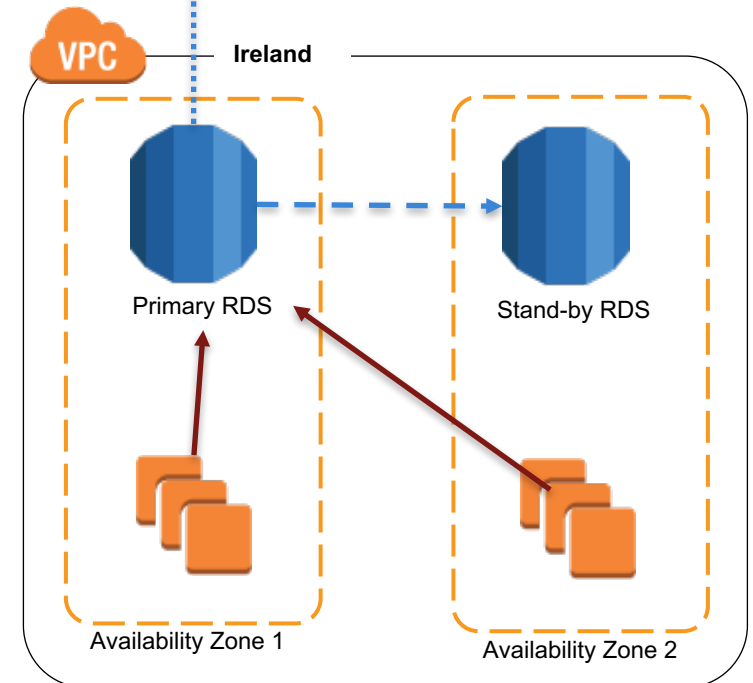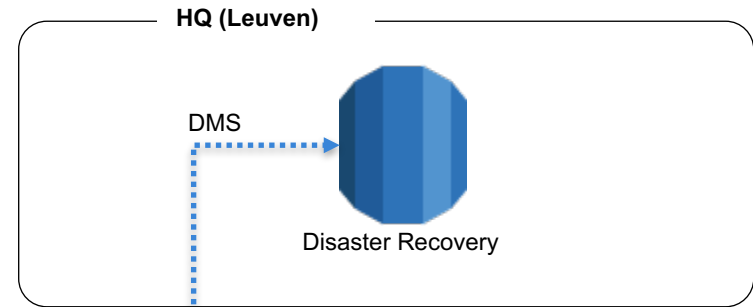- Cloudformation templates
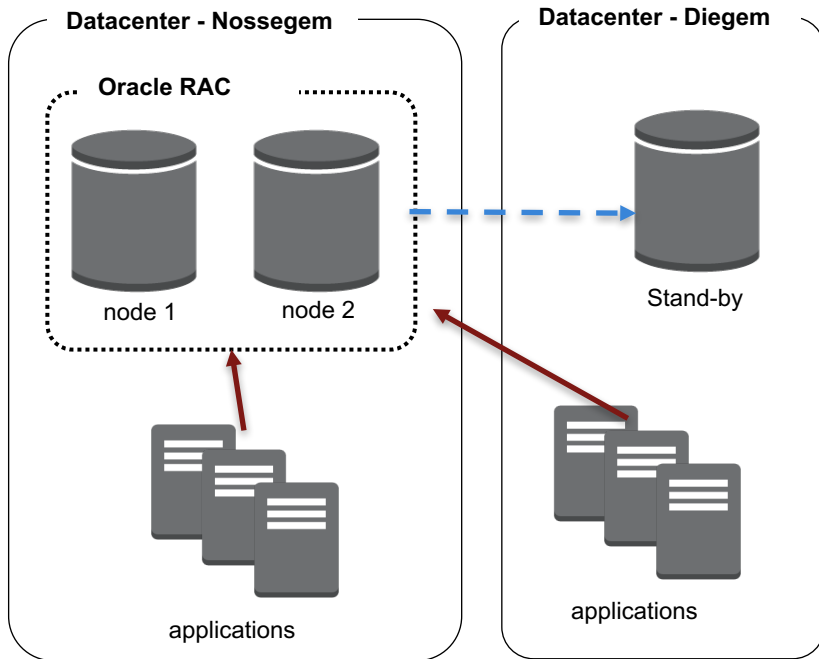
Overview environments
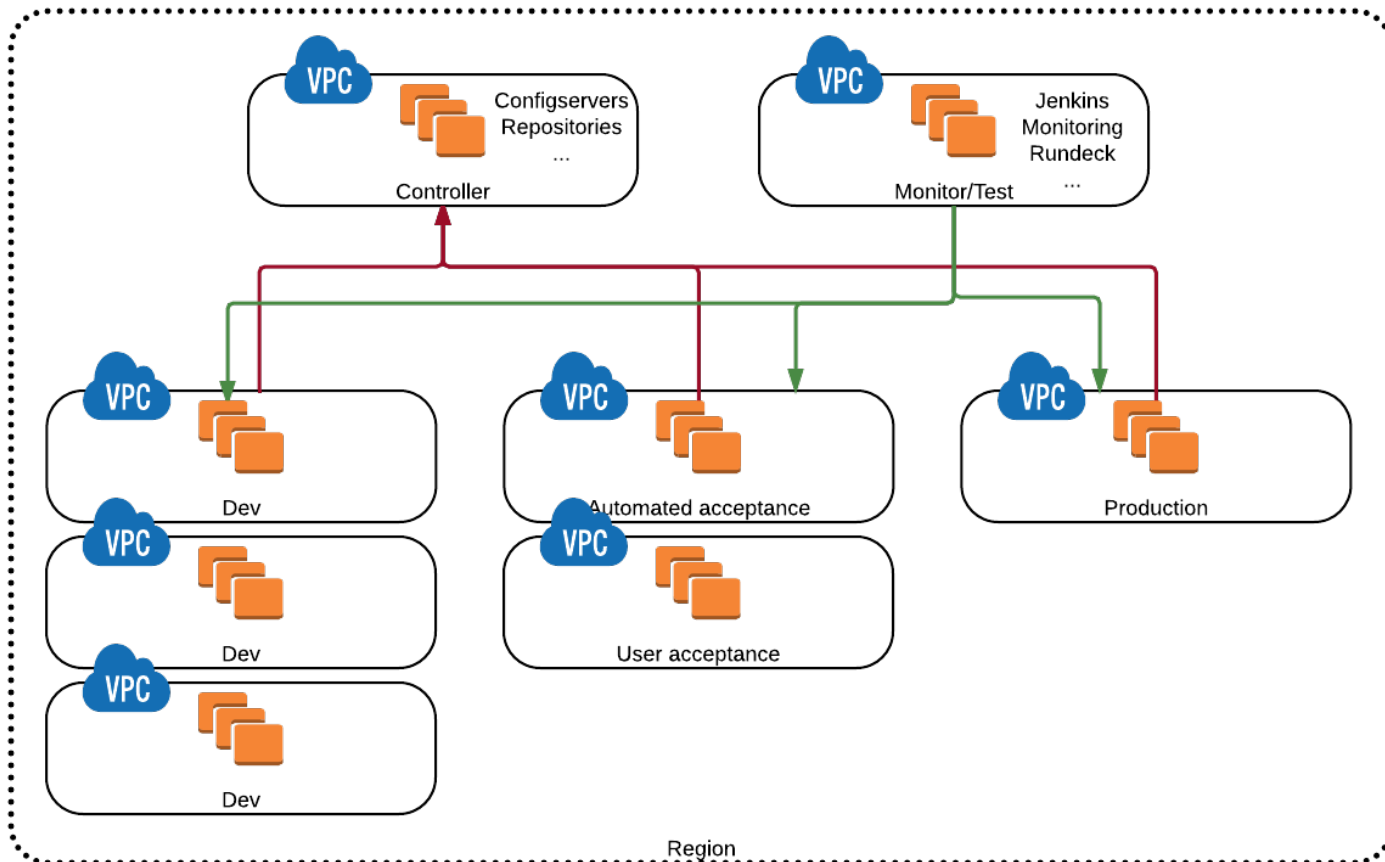
High availability

- All components
    - distributed over 2 availability zones within one AWS Region

    - active-active

    - behind Elastic Load Balancers

    - Intelligent health checks

    - Share content via RDBMS or via EFS (= NFS like)

- All RDBMS instances in multi-AZ mode

Oracle – multi-AZ RDS

- On-prem
  - Both RAC nodes in same DC
  - Manual fail-over to stand-by instance
- AWS: multi-AZ RDS
  - Synchronous Replication
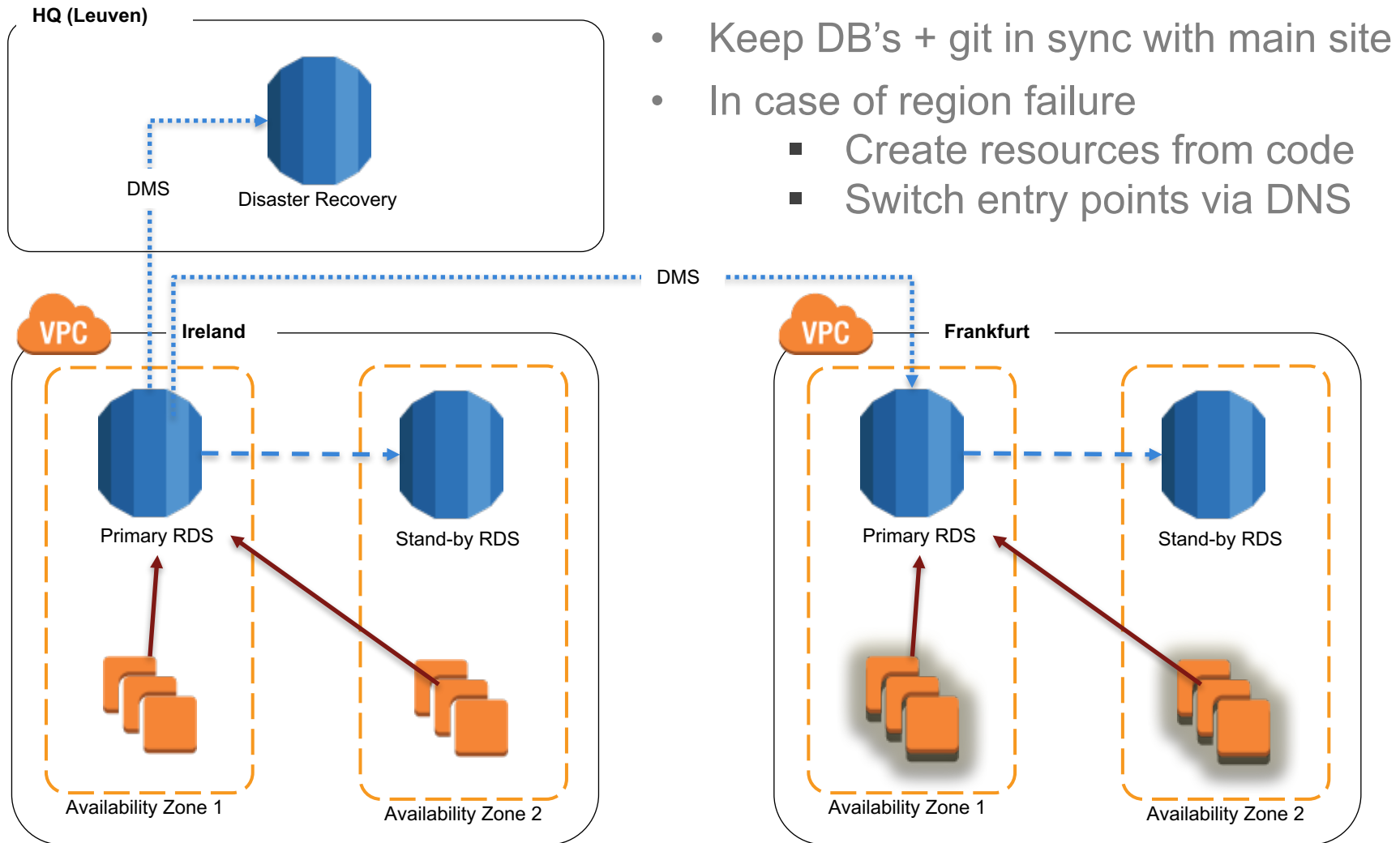 - Automatic & Transparent Fail-Over

RDS & Database migration

- Amazon RDS = enormous time saver !

- No OS level access on Amazon RDS
  => DataGuard etc not an option

- Amazon Database Migration Service (DMS)
  too immature for the migration

- Used complex Oracle Datapump export / import sequence
  instead

- Temporarily up-scaled Oracle instance

- Final export / transfer / import / verify : 2.5h

Experience so far

- RAR's dealt well with change of IP addresses

- Overall satisfied with quality of service & docs

- No performance issues

- Not impacted by S3 outage in US

Next step– Full DR site in another region

- Keep DB's + git in sync with main site
- In case of region failure
  - Create resources from code
  - Switch entry points via DNS

HQ (Leuven)

DMS

Disaster Recovery

VPC

Ireland

DMS

VPC

Frankfurt

Primary RDS

Stand-by RDS

Primary RDS

Stand-by RDS

Availability Zone 1

Availability Zone 2

Availability Zone 1

Availability Zone 2

Next steps

- Disaster Recovery site in another region

- Fully automate Continuous Delivery Pipeline

- Blue / Green deployments

- Nameservers in the cloud ?

- Multi-cloud ?

- Serverless architecture ?

The team