



Dienst Uitvoering Onderwijs  
*Ministerie van Onderwijs, Cultuur en  
Wetenschap*

From 6 weeks to 6 minutes

Jan-Taeke Schuilenga  
IT Infrastructure Architect  
jantaeke.schuilenga@duo.nl



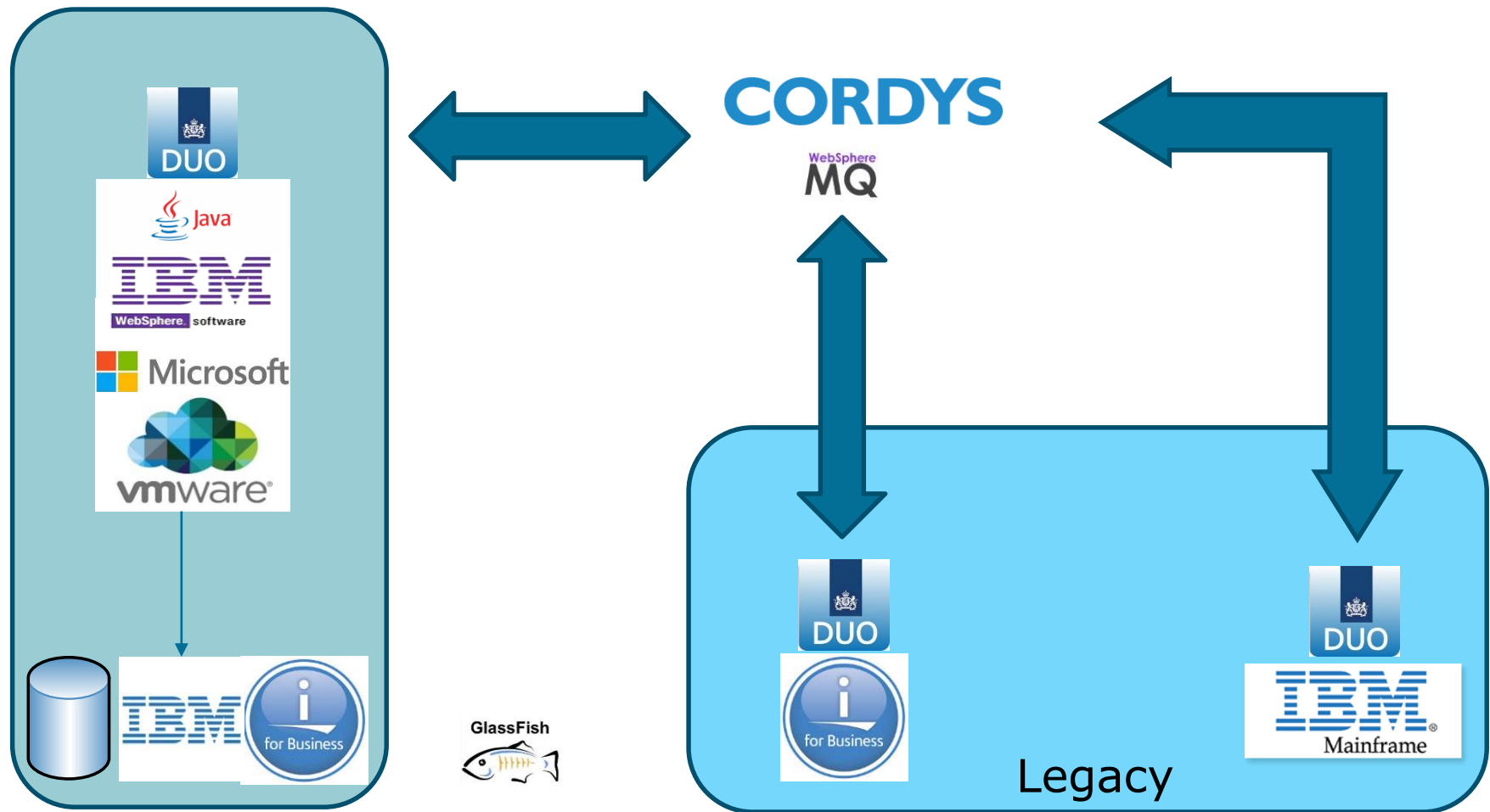
## About DUO

Education Implementation Service of the Ministry of Education, Culture and Science

- Student Finance (€4.3bn)
- School Finance (€30bn)
- Exam services
- Childcare Register
- Government Data Center (ODCN)
  
- 2700 employees, 500+ IT staff



# DUO Infrastructure 2010





## Business not satisfied

- **Quality:** IT products did not meet the business requirements
- **Time-to-market:** IT could not deliver fast enough



## Vision

The processes and the infrastructure for the realisation, testing and management of JAVA applications comply with the principles of **DevOps and Continuous Delivery**



## And there were 2 other problems...

- JAVA 5 to JAVA 6 migration
- Websphere lacks financial scalability



## Selection criteria

### Functionality:

- J2EE Full profile certified
- Management tooling

### Support:

- Enterprise support available

### Market penetration:

- Relevant references

### Financial:

- Financially scalable
- Free version available

### Open Source:

- Preferred, not required (comply or explain)



And the winner is.....





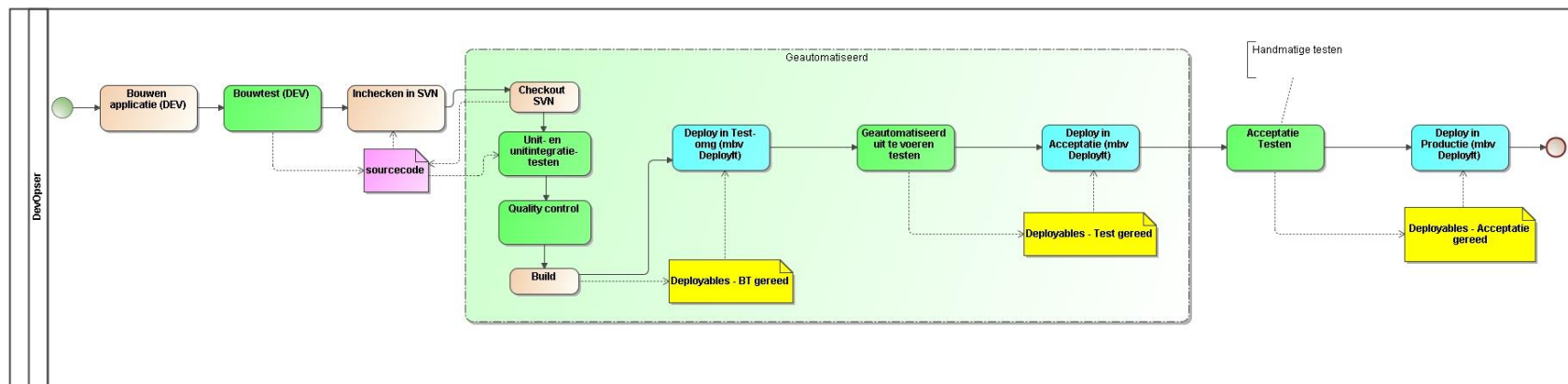
## Xebia Maturity model for CD

	DevOps	Reporting	Testing	Provisioning	Deploying	Building
Level 5 Complete	Operations and development are both part of the multidisciplinary delivery team and share responsibilities.	Reports also provide trend analysis.	100% fully automated tests all the way to production	Self-service portal for requesting an environment.	Continuous end-to-end deployments.	End-to-end automated gated builds.
Level 4 Advanced	An envoy of operations works along in project, an envoy of development works along with operations.	Dashboard provides insight from different perspectives and shows history and progression through a build monitor to all.	Automated dynamic quality tests like security scans, functional and performance tests guarantee quality of code.	Environments can be created and torn down by a push of the button. Operating System is virtualized.	Test-gated deployments of end-to-end applications. Deployments occur over multiple environments.	Central build environment. Teams actively reuse generic components in a secure and controlled manner.
Level 3 Average	Development and operations work together when this is required.	Graphical and textual reports accessible through dashboard.	Automated static code and security analysis after code check in.	Environments are identical. Several tools used to provision and configure an environment.	Environments are identical. Roll out of applications performed by a push of the button. Auto-deployment to D, T, A and P.	Build on commit. Archived components are made available for reuse by other teams.
Level 2 Beginner	Code accompanied with release notes with which operations should install and manage the application.	Reports generated on request by system administrator. Reports are graphical in nature.	Automated tests are initiated as soon as code is checked in. Tests are focused on unit /component testing only.	Scripted installations per component for each server. Surrounding systems manually configured.	Self service deployments to development and test.	Automated builds are performed in a central area and activated manually.
Level 1 Base	Operations engaged at the end of the project.	Reports generated on request by system administrator. Reports are text based.	All tests require manual activity. Some tests are automated but have to be initiated by hand.	Manual installation and configuration of software for middleware, databases, application servers, etc.	Deployment through execution of separate deployment -and db scripts. Manual configurations and installs / env.	Builds are performed on local workstation by use of one or more separate build scripts.





# New process...





## Principles

- We standardize everything.  
Machines, infrastructure, test data, application configuration
- We automate everything.  
Provisioning, Deploy, Test
- We use knowledge and expertise from partners (RedHat, Xebia)
- Self service  
Developers can create, delete and change their own infrastructure  
Operations can (un)deploy their applications to all environments  
Developers can monitor their applications and infrastructure

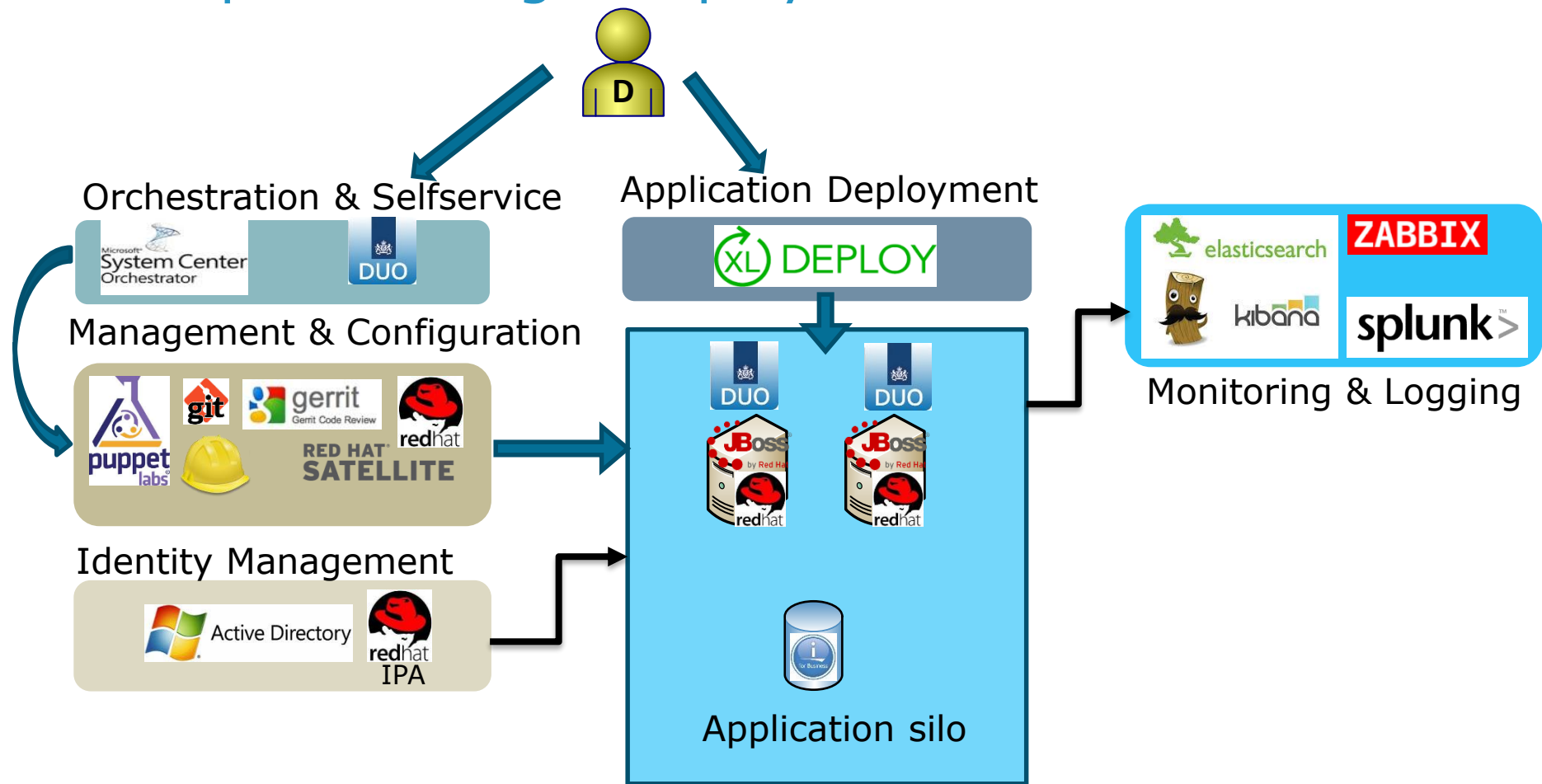


## Silo

- A logical application container which contains several generic infrastructure components
- Currently supported components
  - *Pre-configured and hardened (security) JBoss EAP 6 instances*
    - *resource adapters, data sources ,drivers and default ports*
  - *DB2 databases & schema(s)*
  - *coming up...*
    - *Fuse ESB*
    - *WebSphere MQ Queues and Topics*
    - *etc.*

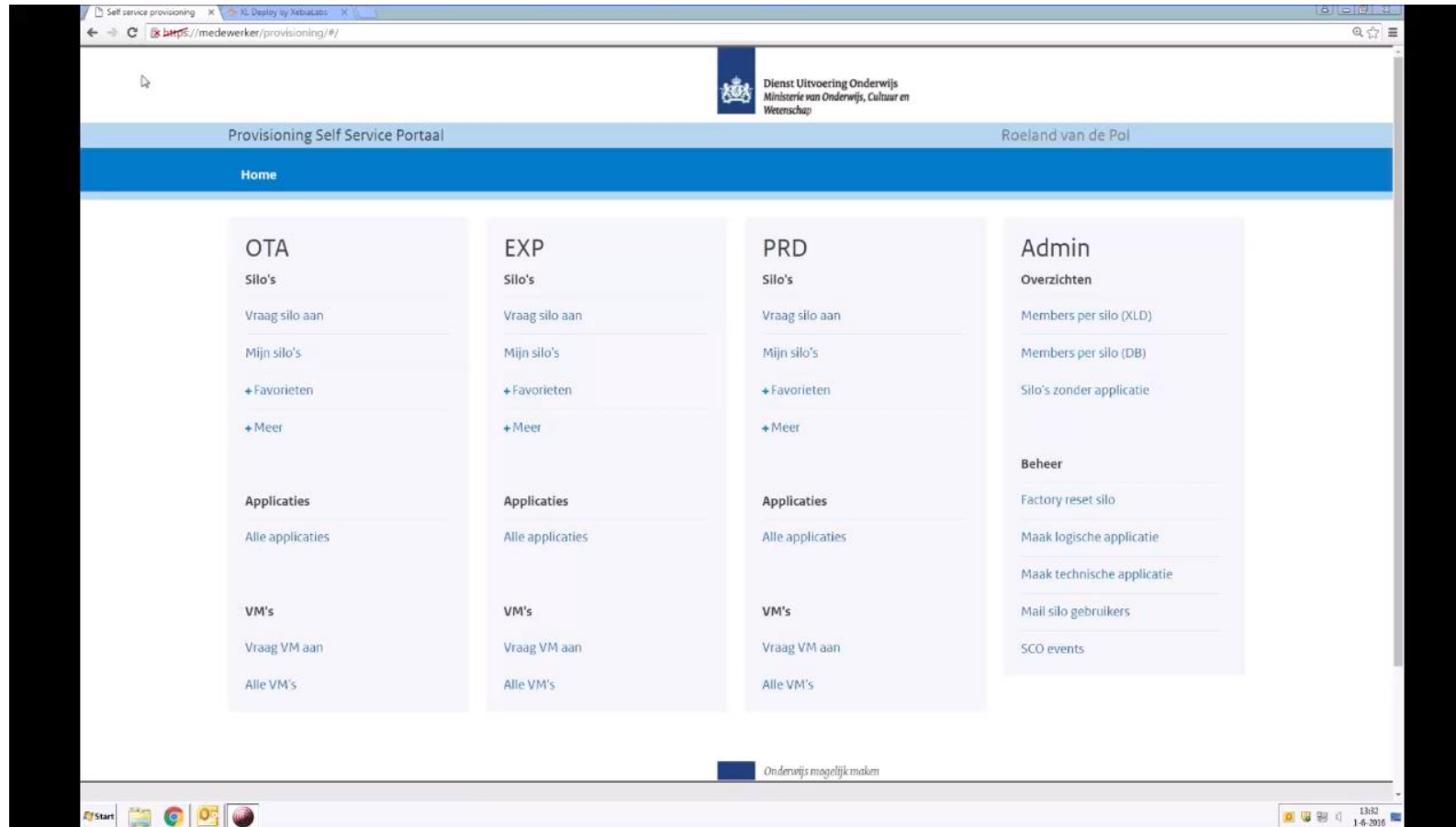


# DUO provisioning & deployment infrastructure





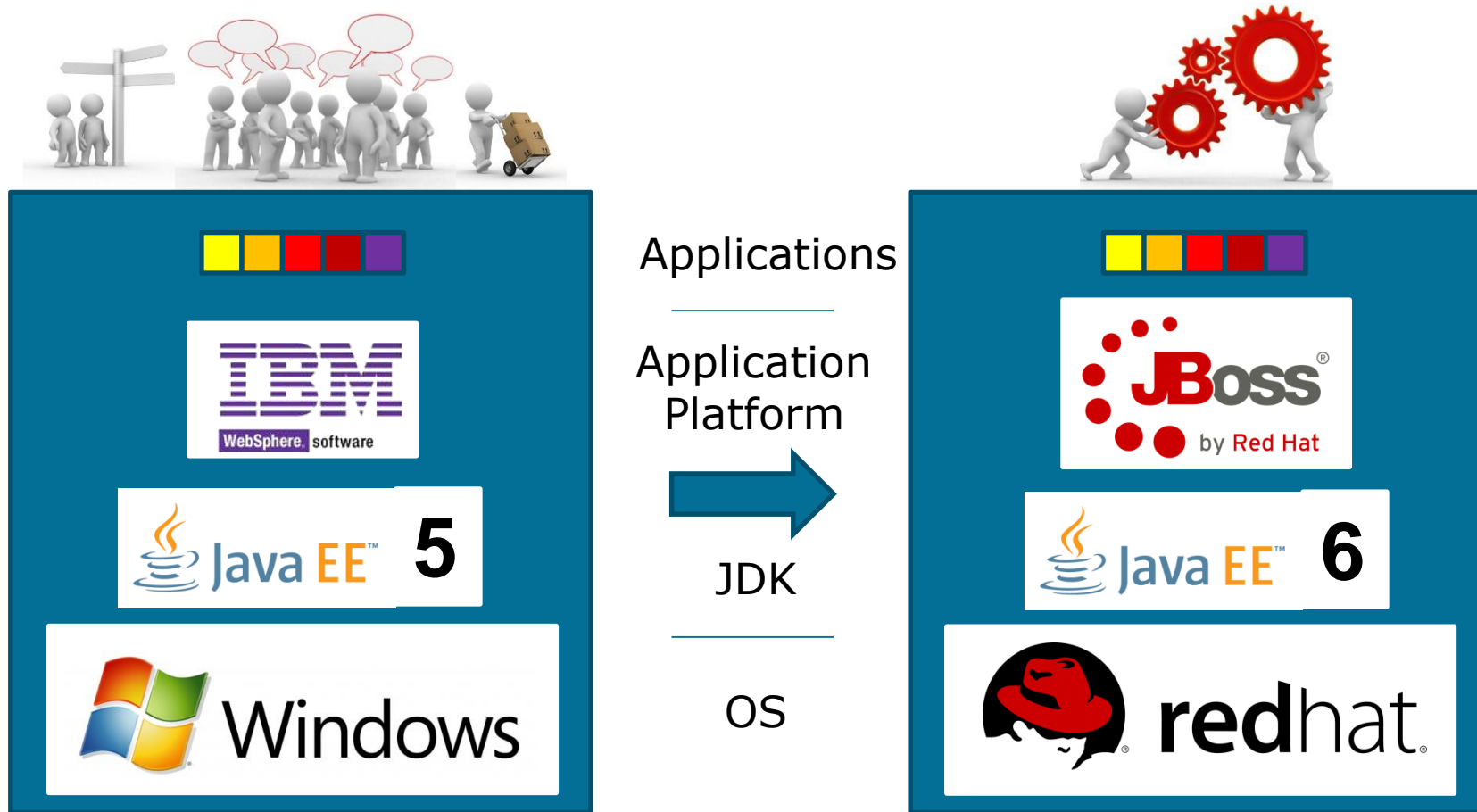
# Demo



<https://youtu.be/yalAqIxQUaw>



# Application migration





## Migration Phase 1- Preparation (2014)

- Migration assessment
- Proof of Concepts
- Pilot projects
- Planning
- Documentation



## Knowledge management

- Knowledge sharing crucial!
- Efficient, motto: "No problem is solved twice, no question is asked twice."
- How? central knowledge library
  - Step-by-step Migration Manual
  - Migration Cookbook
  - Platform FAQ
  - Pilot projects
  - Core team



## Migration Phase 2- Large scale migration

- Application migration
- Enablement workshops
- Support teams and applications
- Improvement cycle: system, standards, documentation ...

2016 Q4: IBM Websphere decommissioned



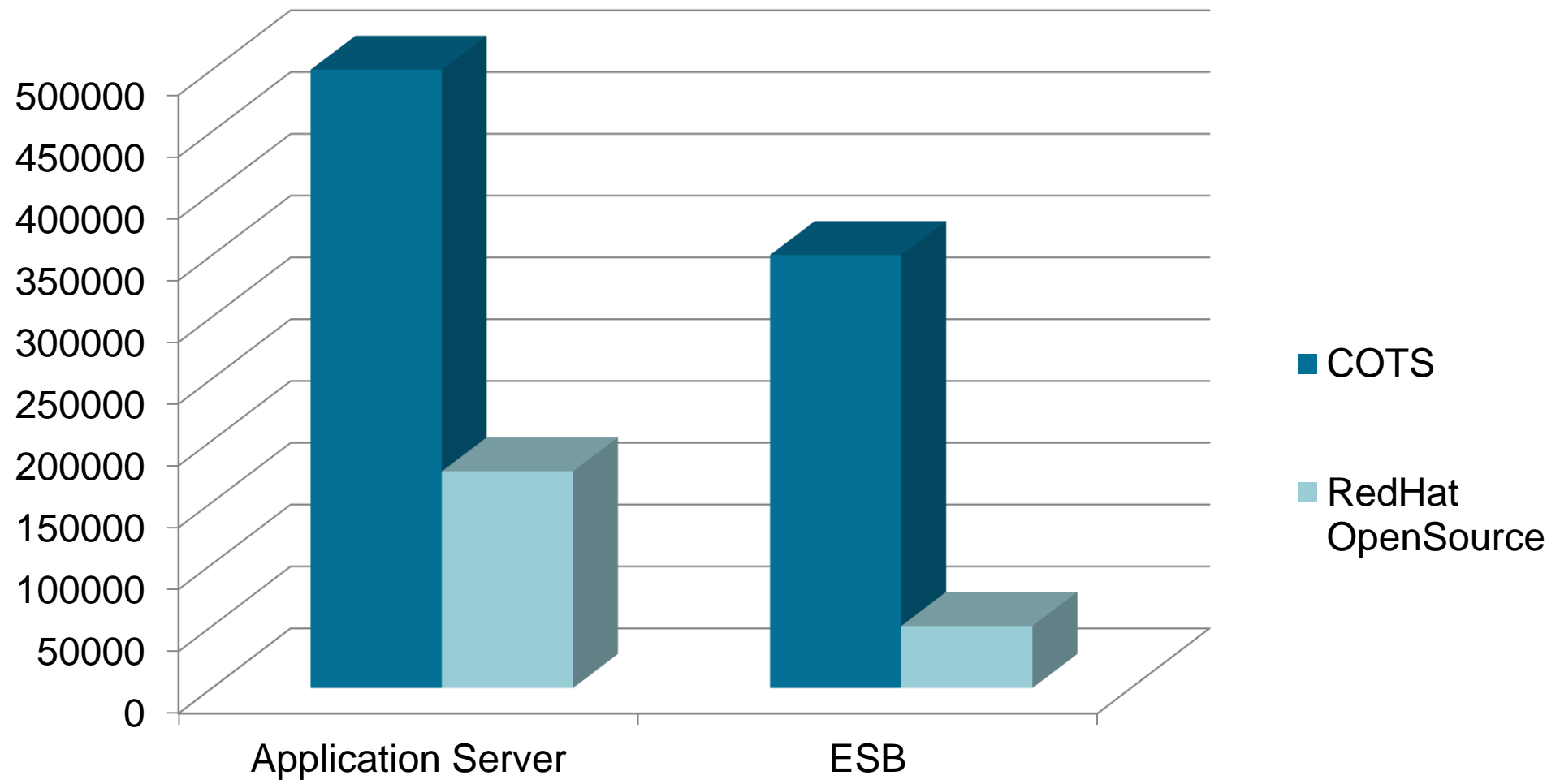
## Findings

- OpenSource functional/technical equivalent to other commercial enterprise products
- More flexibility
- Fast provisioning and deployment.
- Developers and Business have become enthusiastic
- Financially scalable



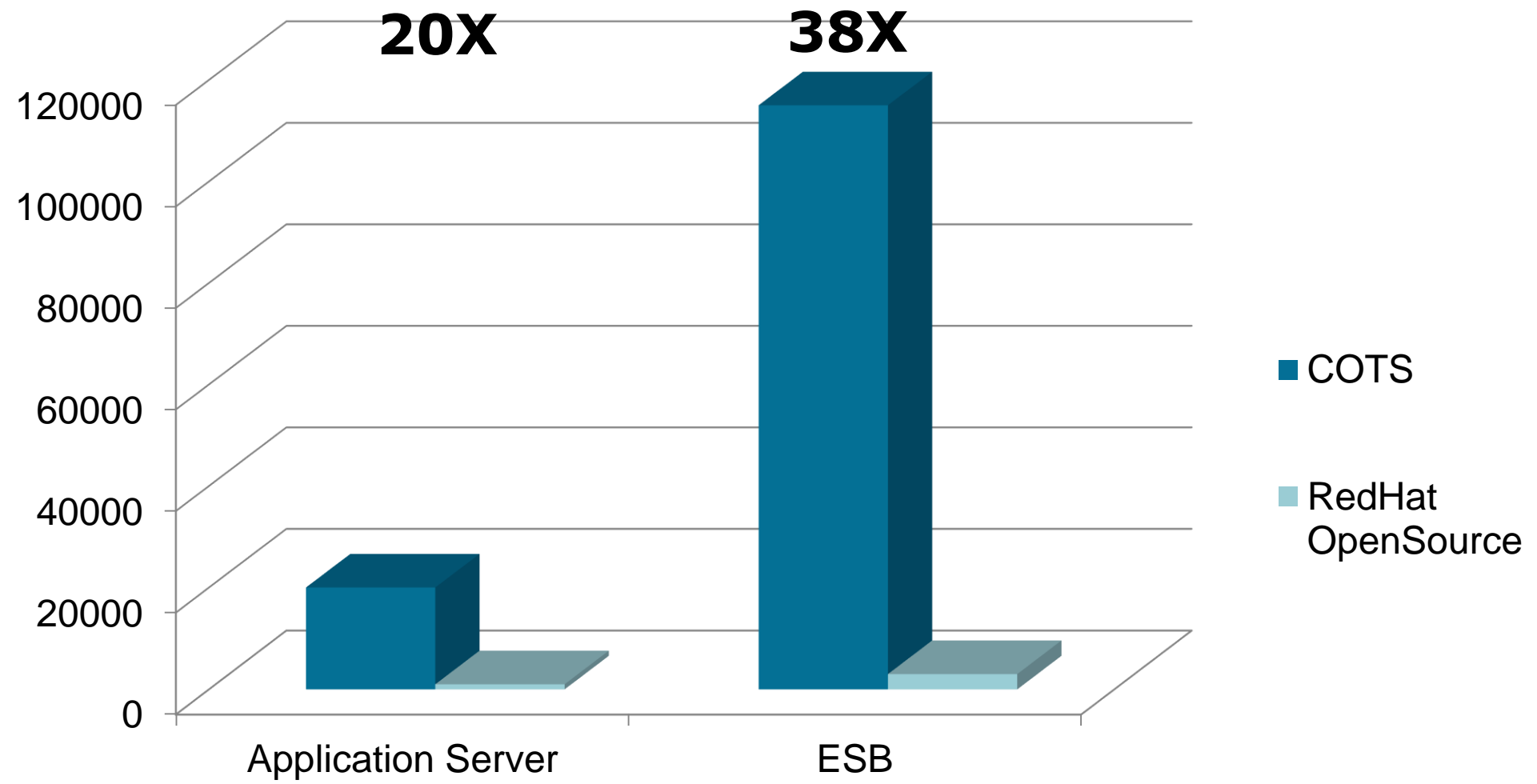
# Annual Software Support **3X**

# **7X**





# Financial scalability: 2 extra cores will cost you....





## Roadmap

- 2016/2017 Replace Cordys ESB with RedHat Fuse
- 2017 PaaS/ Containerization (OpenShift?)
- 2016/2017 DevOps organization (40 teams)



# Questions?

