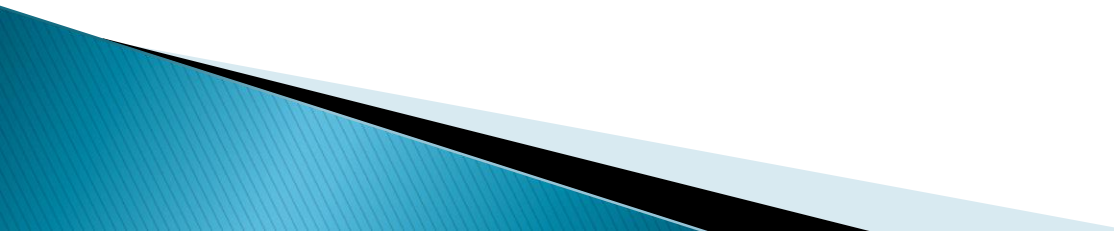# Pas de deux: a tale of test automation and service virtualization

Bas Dijkstra, november 2014

# Outline

- Introduction
  - What is service virtualization?

- Case study
  - Challenges
  - Solutions

- Benefits of service virtualization to testing and test automation

- Configuring virtual test environments from within the automated testing solution

About me

- Freelance test consultant
  - The Future Group

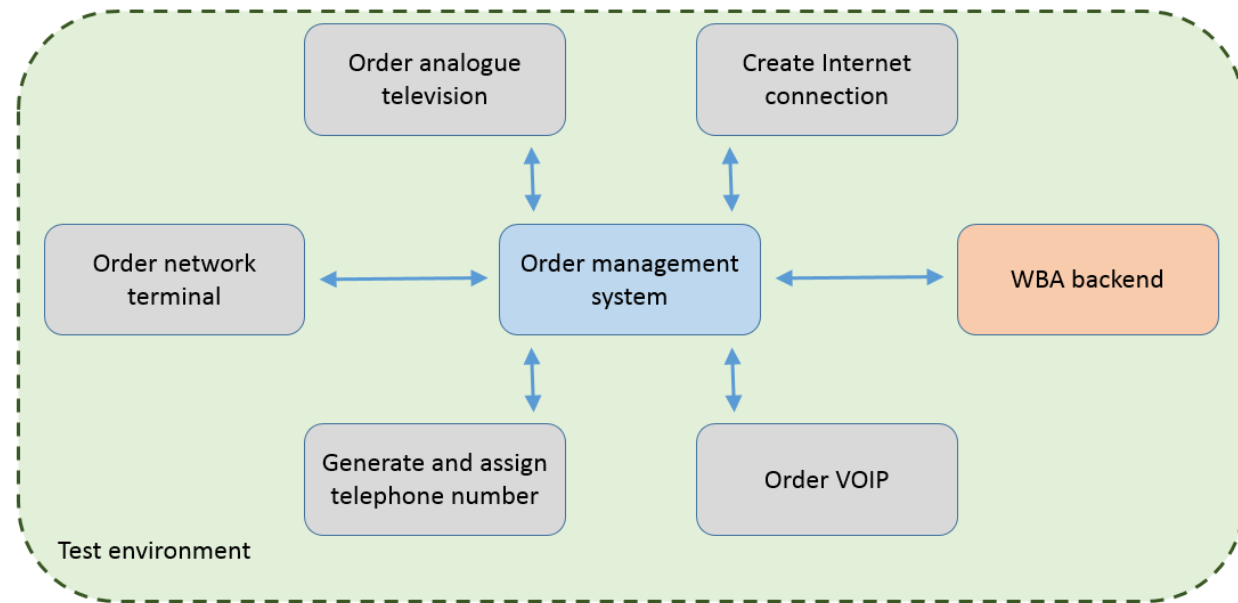- 8+ years of experience
  - Test automation
  - Service virtualization

What is service virtualization?
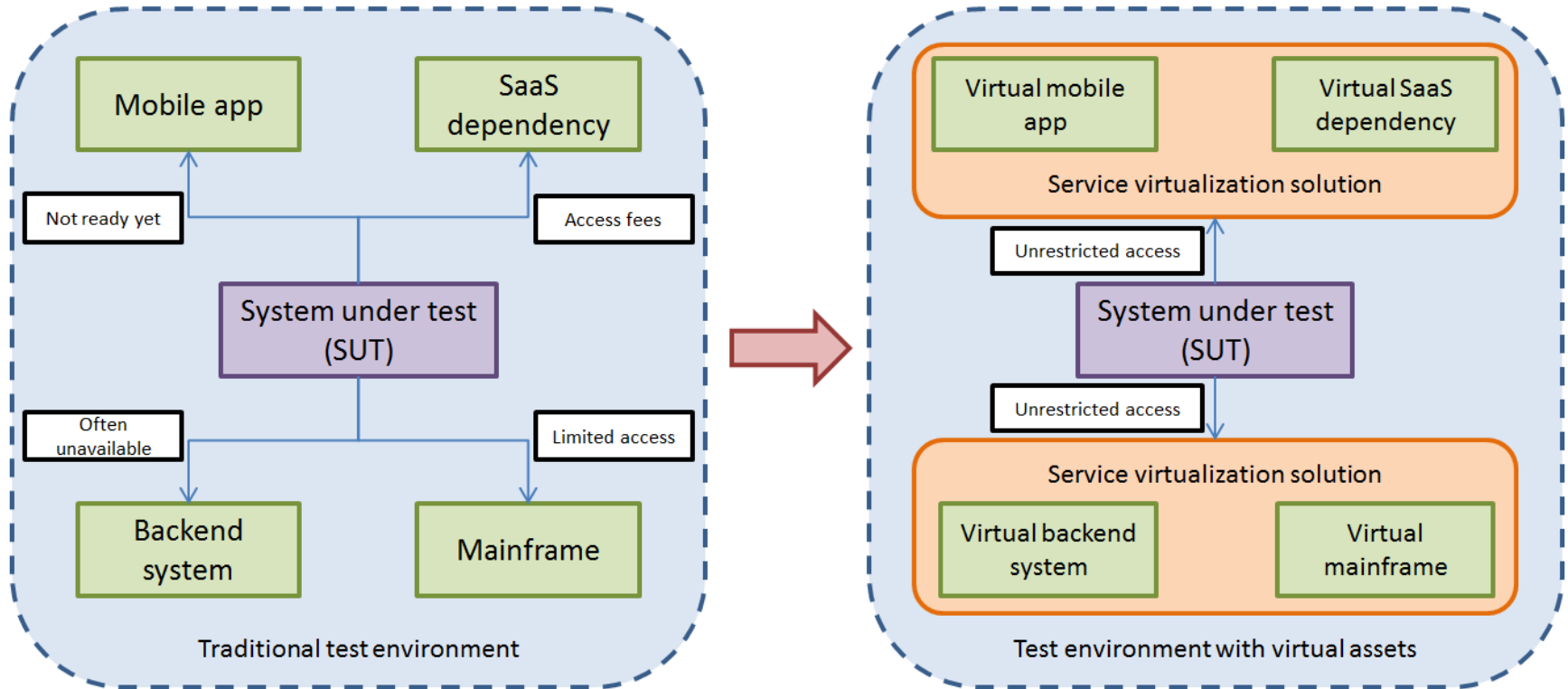
Simulation of dependency **behaviour**

Case study

- Large provider of telecommunication services in the Netherlands

- Order management system
 - Typical provisioning action touches around 10 dependencies

Challenges

- Order data needs to be present in back-end system

- Requires manual configuration of order
  - This could take days (or even weeks...)

- As a result:
  - Very limited test data
  - Long test cycles
  - Low test coverage
  - No test automation

Solutions

- Introduction of service virtualization

- Simulation of back-end behaviour
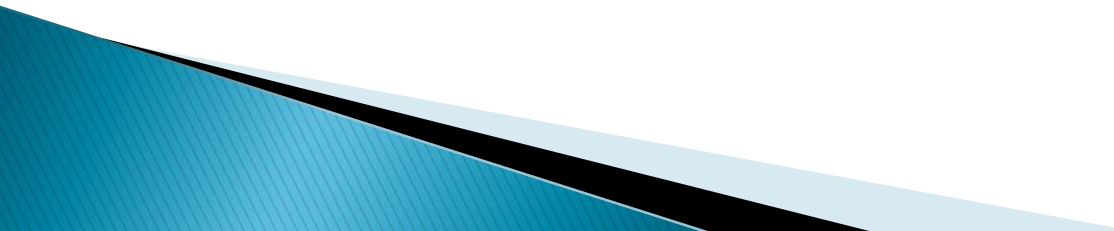
- Virtual asset acts as if order is present
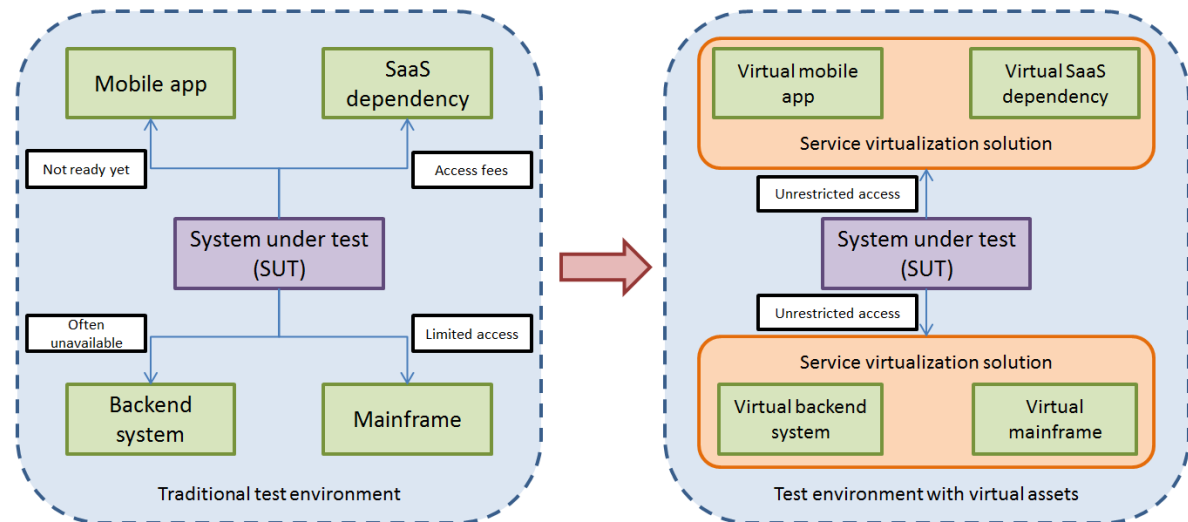
Results

- Order provisioning takes minutes, not weeks

- Test team has 'unlimited' test data

- Tests can be automated

- Organization can make the move to Agile

Benefits of service virtualization for (automated) testing

- Enabler for test automation

- Faster testing

- More control over test data

- Increase of test scope

Enabler for test automation

▸ Previously impossible to automate tests

▸ Reduction in time is enabler for automated tests
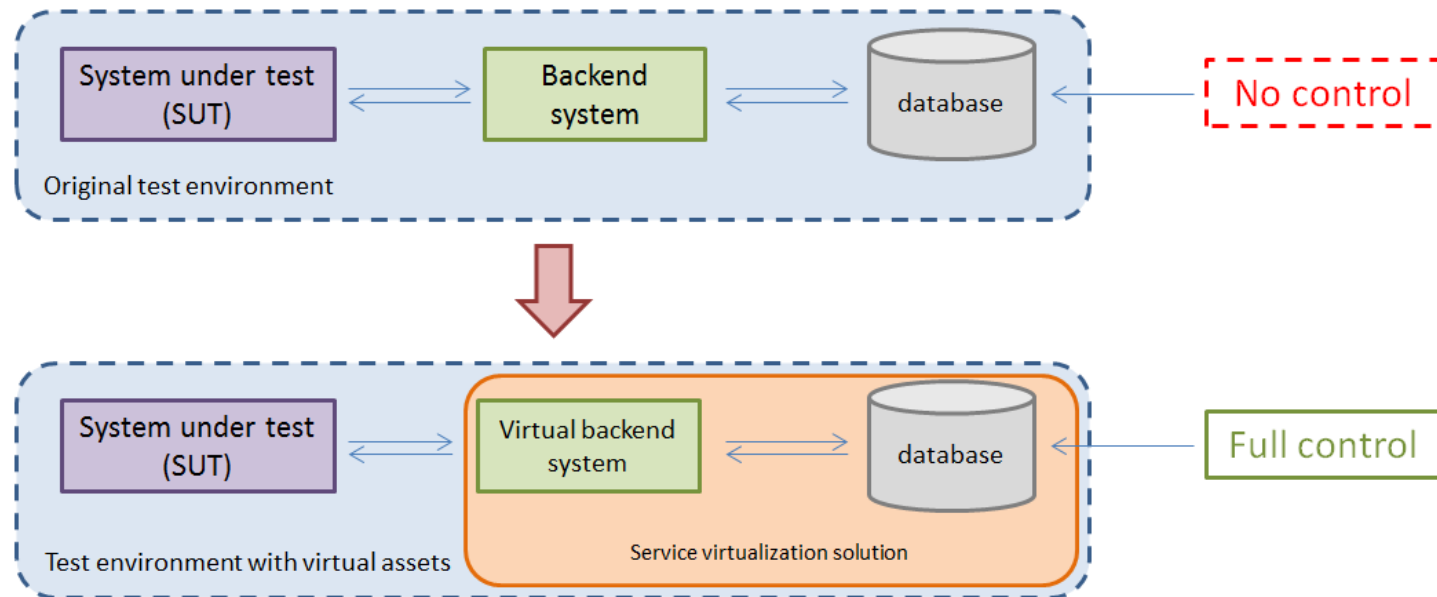
▸ No more manual regression testing

Faster testing

- No more need for weeks of testing

- Developers can do integration testing
  - "Shift-left" of testing process
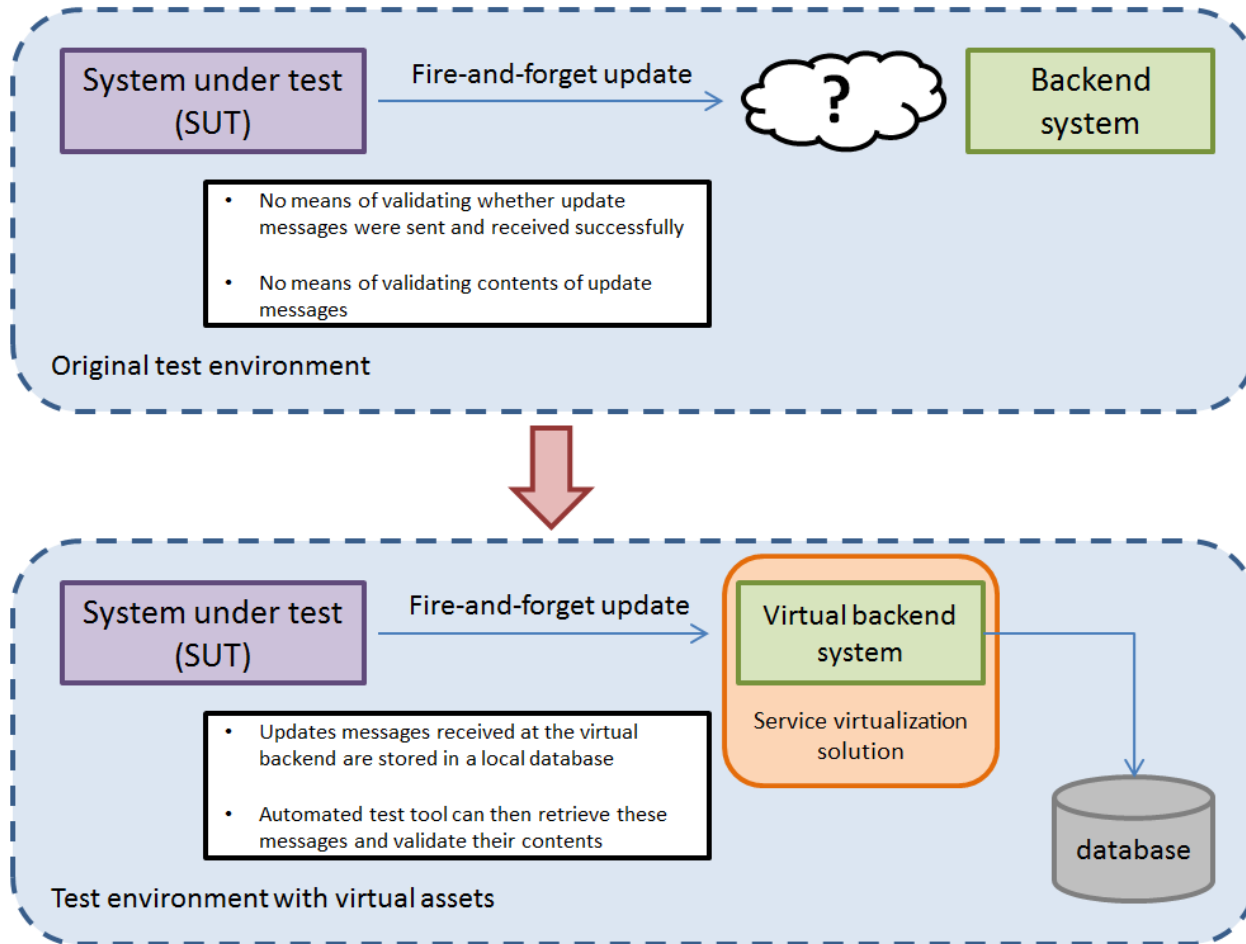
- Bug fixes can be retested on the spot

More control over test data

- Testers can add and edit test data

- Easier to test specific combinations and situations
  - Increase in test coverage

Increase of test scope

System under test (SUT) — Fire-and-forget update → ? **Backend system**

- No means of validating whether update messages were sent and received successfully
- No means of validating contents of update messages

Original test environment

System under test (SUT) — Fire-and-forget update → **Virtual backend system**

Service virtualization solution

- Updates messages received at the virtual backend are stored in a local database
- Automated test tool can then retrieve these messages and validate their contents

database

Test environment with virtual assets

Automatic configuration of virtual test environments
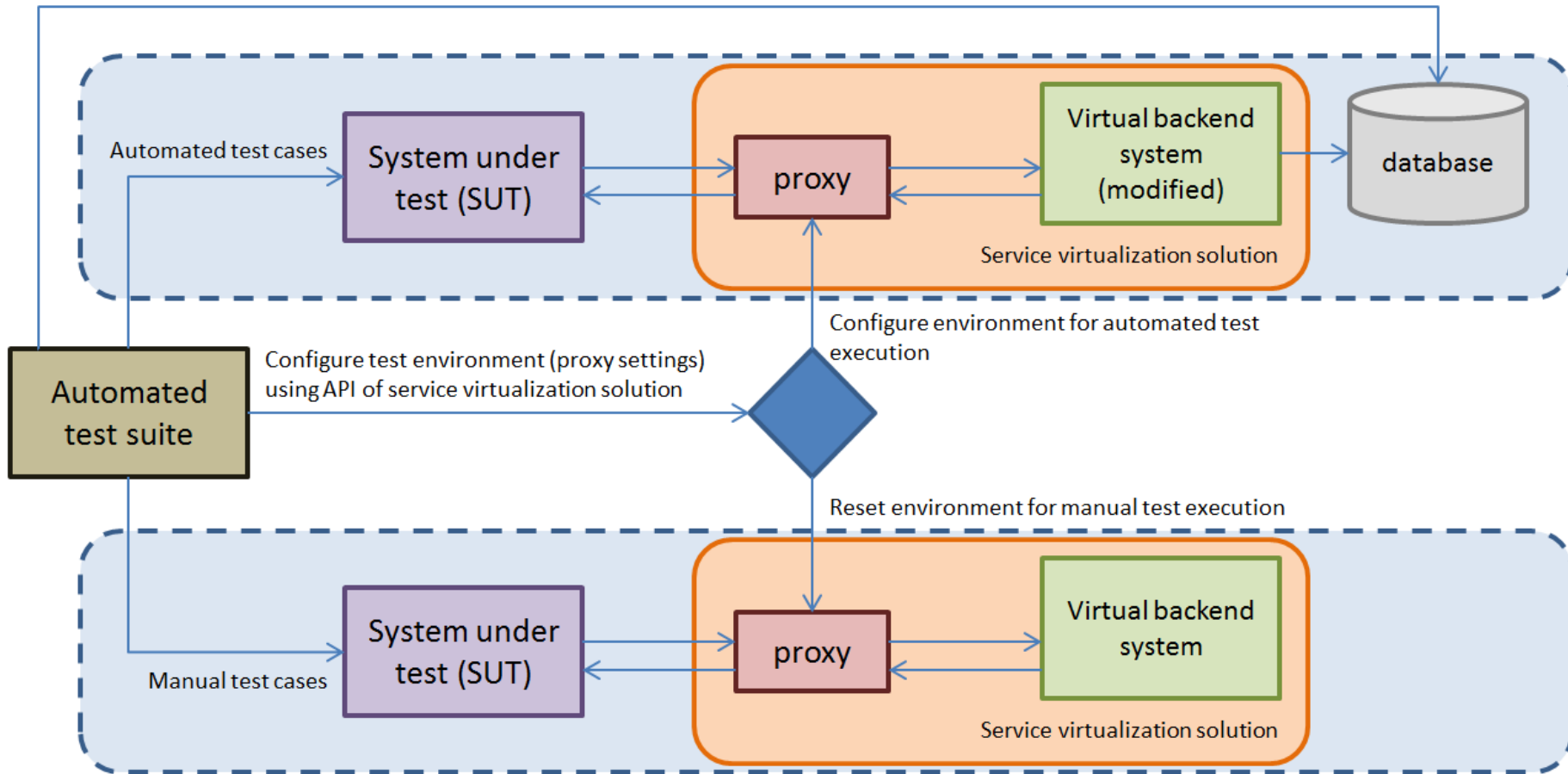
Automatic configuration of virtual test environments

Execute validations on database level

Automated test cases

System under test (SUT)

proxy

Virtual backend system (modified)

database

Service virtualization solution

Configure environment for automated test execution

Automated test suite

Configure test environment (proxy settings) using API of service virtualization solution

Reset environment for manual test execution

Manual test cases

System under test (SUT)

proxy

Virtual backend system

Service virtualization solution

Questions?

- basdijkstra@gmail.com
- LinkedIn