Pull Deployment of Services
Introduction, Progress and Challenges

Sander van der Burg

Delft University of Technology, EEMCS,
Department of Software Technology

Philips Healthcare, Philips Informatics Infrastructure (PII),
Best

December 11, 2009
Hospital environments

- In hospitals many tasks are performed which require access to data e.g. images, patient records, measurements
- A hospital infrastructure is currently *device-oriented*.
  - Services are assigned to fixed devices
Hospital environments

Implications:

- Resources are assigned to services, *even when not used*
- *Inflexibility* in reacting to events (e.g. a device could be broken)
- A stakeholder has to go to a particular device to perform a certain task
Hospitals are complex organizations:
Service orientation

User → Service
Service orientation

User → Workstation → Medical data
Service orientation

User → Workstation → Medical data

User → Mobile phone → Datacenter
Service orientation

Sander van der Burg

Pull Deployment of Services
Goal of PDS is realising an *architecture*:

- Designing and implementing distributable services
- Distributed (re)deployment of services
PDS architecture

Service design layer

Deployment layer

Infrastructure layer
PDS architecture (progress)

- Service design layer
- Deployment layer
- Infrastructure layer

Pull Deployment of Services
Disnix

- A distributed software deployment tool
- Built upon Nix (TraCE project)
- Allows users to capture components of a distributed system, the network and the mapping of components to machines in models
- Automatically deploys the specified components into the network
Tested on the Service Development Support System (SDS2) developed by Philips Research
Pull Deployment of Services
A user can specify a network of machines in a model.
The model can be used to automatically deploy the machines in a network.
The model can be used to generate virtual machines and perform testcases.
Challenges

- Disnix has to be extended with more features
- Performing more case studies:
  - Philips case studies
  - Open Source case studies
  - WebDSL (MoDSE project)
- More experimenting with testing techniques (cooperation with ARTOSC project)
Platform used to build Philips medical applications

- Very large case study (large platform, many components, legacy software)
- Built upon the .NET platform
- Our tools need to be ported
PDS architecture (challenges)
PDS architecture (challenges)
Cooperation with Philips

- Philips offered me a 3.5-day PII introduction course
- Access to PII platform source code
- I work 1-2 days at Philips every week
- Currently, working on support an initial subset of PII with our tooling
  - PII environment is an “alien” environment for our tooling
- Future work:
  - Communicating with PII developers in supporting larger subset
  - Implementing/integrating other parts of the PDS architecture
We have realized *Disnix*, a distributed deployment tool and applied it on SDS2

We have developed a technique to test distributed systems

Lots of challenges remain to be done in realising a PDS architecture
Our tools are released as Open Source software:

- Nix, NixOS, Disnix: http://www.nixos.org
- WebDSL, http://www.webdsl.org