Motivation

In hospital environments various services are used to assist medical staff in performing their tasks, such as retrieving patient records or viewing X-Ray images. Such environments are mostly device-oriented. For example, in order to view an X-Ray image, a doctor must use a dedicated workstation assigned for this purpose.

Device-orientation has several undesirable implications:

- **Overcapacity and suboptimal usage.** System resources are only used while accessing a service on a device.
- **Inflexibility** in reacting to events. If a device breaks, a service may become completely inaccessible.
- **Deployment** of services is a complicated and time-consuming process, because deploying components is performed in a semi-automatic and ad-hoc fashion.

Goal

Hospitals are moving from device-orientation to service-orientation: services should be accessible from any device and any place. This requires that the software components that implement a service can be deployed automatically to the devices where they are needed (pull deployment of services). Since some devices may have limited resources (e.g., iPads), it may be necessary to deploy some components to other devices in the networks that do have sufficient resources.

The goal of the PDS project is to realise an architecture for the design of distributable services and tools to deploy such services automatically in complex, heterogeneous environments. While PDS is motivated by the medical domain, these techniques can be applied in any domain that uses service-oriented systems.

Results

**Service deployment**  We have developed Disnix [1, 4], a system for automatic deployment of distributed services.

**Infrastructure deployment and system testing**  We have developed techniques for modeling and deploying networks of (virtual) machines on the basis of declarative specifications. Such models have various use cases [3], such as automatically deploying physical or virtual machines (for example in a cloud infrastructure), or running automated tests of distributed systems with complex environmental dependencies [2].

Participants

**References**


