Requirements for a Software-Intensive Ecosystem for Telemedicine

By Christensen, Christensen, Hansen, Kyng, Manikas, Surrow, and Urazimbetova

Med-e-Tel, Luxembourg, May 2012

Speaker: Henrik Bærbak Christensen
Aarhus University
Net4Care sponsors and participants...
The Challenges

Denmark faces

– Demographic changes: More old people, less young
  • In 2044: 60% more elderly people than today
  • (And Denmark has high birth rate: 1.8 child pr women)
– Life-style related chronic diseases increase
  • In 2009: Estimated 80% of public health expenditure

Similar challenges in other developed countries

Part of a solution: Telemedicine

– ”Out-patient clinic” in the home
  • Measurement upload, video conferencing
Net4Care: The Vision

Observations
  – Many experiments but...
    • End-to-end / Silo solutions...
    • Too expensive, too complicated, too broad
    • Abandoned when project money ceases

Vision
  – Software Ecosystem for Telemedicine
  – Open source reference implementation
  – Modular, reusable, flexible
  – SMB marketplace
Software Ecosystems

Working definition
- A software ecosystem consists of a set of actors and software elements in a domain interacting as a system

Some examples
- Microsoft Office
- Google Android, AppEngine

Contrast
- Product Lines (single vendor)
Easy at first look...

Many pieces apparently already there...

– HL7 v2 and v3
  • Formats for messaging and clinical documents

– IHE profiles
  • XDS: Cross Enterprise Document Sharing
  • Repository and Registry for storage

– MedCom standards and web services
  • Danish successful initiative
(Very) steep learning curve
  – HL7 documentation overwhelming but *reference type*
    • Requires you to know it before it really makes sense
  – Examples and code related artefacts hard to find

Only proprietary (expensive) environments
  – Prohibitively high initial cost
  – Not appropriate for research/education institutions

Either no or transient testing environment
  – No sandbox for testing and experimentation
  – “Code working today does not work tomorrow”
Initial focus

Clinical observations in the home / monitoring
- Diabetes, COPD, blood pressure, ...

Application-centric Ecosystems
- Simplify contribution
- Extends data models

Success factors

The following success factors can be identified for this category:

- The foremost success factor for any application-centric software ecosystem is a large set of customers or the promise of those customers that have real reason to extend the platform with additional functionality. Although the “coolness” factor certainly plays a role, third party developers are driven by their business case, i.e. a pool of potential customers.
- Assuming the first factor is met, the platform company should aim to simplify contributing by third party developers through allowing the use of generic, popular development environments, stable and expressive interfaces and, in the case of web applications, easy deployment and integration with the platform.
- Although most application platform companies focus on providing access to data, providing solutions to extend data models and workflows as well as integrate in the same user experience framework is important to achieve a seamless integration from a customer’s perspective.
- Finally, third party developers need paying customers and the platform company needs to take a role in providing a viable channel where 3rd party contributions are exposed to customers.
Business Scenario

Developer in SMB wants to develop spirometry application for the home
- Measure spirometry data by easy user interface
- Upload measured FVC and FEV1 to national XDS servers for access by all regional EHR and GP systems

1st prototype of such system running in 4 hours!
Net4Care Ecosystem

Architectures overview
- Tutorials
- Learning
  - Source code examples

Information
- Reference
  - API documentation
  - Protocol documentation
  - Dataformat documentation
- Configuration consistency
  - FAQs
- Community
  - Support
  - Bug tracking
- Module catalogue

Architecture
- Core deployment unit
- Boilerplate generator

Protocols
- On the Wire handshake

Dataformat
- Persistent format
- Exchange format

Components
- Functionality
- Security
- Device drivers

Authentication
- Auditing
- Human communication

Search/Registry
- Storage
- Authentication service
- Event system
- Bridges

Services
- Fake objects for services
- Virtual environment

Processes
- Test environment

Certification
- of modules
- of device driver
- of service access
- of data storage

www.net4care.org
A very initial version of the framework is available

- http://www.net4care.org
- (Some) Tutorials with demo code
- Open source implementation
- Staged testing environment

Validation and further development on-going 😊
First ‘organisms’ are moving into the ecosystem