openEHR
Services architecture

Thomas Beale
The Services Part
General approach

- Build from bottom up – ‘Native’ services
  - Needed services, consistent with information and model artefacts
- And from top-down – ‘Standard’ services
  - Connect IHE, HSSP etc to native services
On Services...

- Some architectural inspiration from Microsoft Connected Health Framework (CHF)

- Native layer
  - Take as much of IHE, HSSP etc interfaces & adapt for native compatibility, making *archetype-aware*

- Standards layer
  - Service – service connection
‘Standard’ services
IHE basic services

- Initial services to enable basic info sharing:
  - PIX – person index
  - PDQ – person demographic query
  - XDS – x-enterprise document sharing

- Doesn’t give you semantics, but gets everyone sharing in the same way, and gets everyone to agree on
  - Basic demographic attributes
  - Basic clinical document meta-data items
Value obtained

- Relatively quick to implement
- Enables useful queries like ‘where is all the info on this patient xxxx?’
- Clinicians who are used to paper will have no problems
Going further

- If we want shared structured information, distributed querying, merged medication lists, decision support, public health analytics etc....

- Then we need semantics
  - archetypes

- This means implemented native services that provide access to real underlying information
‘Native’ services
Microsoft CHF

- Process Models
- Service Models
- Information Models
EHR Service

Workflow & Pathway Services

Service Interface Components

- Patient Identity & Health Status
- Patient Consents
- Patient Records
- Patient Management
- Assessments & Care Plans
- Clinical Processes
- Care Pathways
- Professional Roles & Teams
- Professional Permissions
- Orders & Results
- Waiting Lists & Schedules
- Clinical Knowledge Management
- Clinical Decision Support
- Health Classifications
- Clinical Data Management
- Rules Engine
- Healthcare Knowledge Management

Business Components

demographics

Security & privacy

openEHR

© Thomas Beale 2011
The openEHR services architecture

http://www.openehr.org/wiki/display/spec/openEHR+Service+Model

© Thomas Beale 2011
Native services

- All services archetype/template aware
- Numerous APIs generated from templates
- Query service based on AQL / a-path
  - No SQL queries against physical database!!!
Key data services - EHR

- ARCHETYPE-AWARE
- Virtual EHR – fine-grained creation, modification, retrieval, querying
- EHR back-end – coarse-grained DVCS-like interface – ‘change-set’ based
- EHR audit log
Key data services - patient

- Demographics – ARCHETYPE-AWARE
  - Authentication info
  - Patient relationships
  - HCP relationships – teams etc

- EHR subject X-ref service
  - openEHR EHRs are identified by EHR id only
  - Deals with merged & split EHRs, i.e. 2 subject ids ➔ 1 EHR, 2 EHR ids ➔ 1 subject
  - Enables dynamic distribution of EHRs
  - Becoming the EHR meta-data service
Key knowledge services

- Archetypes & templates
- Terminology ref-sets
- Terminology service
  - access
  - Inferencing
  - Terminology administration
- Medications, devices
- Allergies & interactions database
Key process services

- Event-based notifications
- Care pathway
  - Based on archetyped openEHR structures
- Booking / appointments
  - Requires access to patient requests & doctor’s diary & other resource availability data
- Doctor’s diary
  - Forces syncing of appointments to filler
- Patient diary
  - Allows multi-function visits
Key elements that MUST WORK

- Standardised **querying** of data, based on knowledge artefacts, not physical DB
  - → standardised knowledge artefact identification, including versions
  - → standardised ability to designate finest grain items in the data
- Enabling URI to any data item
Services lessons

- Need native and standard layers, i.e.
  - Inside-out, outside-in
- Knowledge-based architecture brings new needs:
  - New knowledge services – archetypes, ref-sets
  - Other services must be knowledge-aware
- Business services need to be small, with changeable interfaces
Knowledge awareness means...

- Service layer understands:
  - knowledge artefact identification system
  - Fine-grained data item identification
- Which means we need standardised knowledge models
- Aka Detailed Clinical Models