

openEHR as the Health Computing Platform for the EHR



Dr Hugh Leslie

Acknowledgement:
Thomas Beale



What is an Electronic Health Record?

Its **NOT** the application
Its the
INFORMATION



openEHR_{1.2}

The EHR Architecture

Problems that an EHR architecture needs to address

- **Semantic interoperability:** how do *different pieces of software* know what the data mean?
- **Patient-centric view:** how to build a patient-centric longitudinal EHR across enterprises?
 - For decision support, Care pathways, Medical research
- **Continual change and complexity:** how to build systems that keep up with reality?



openEHR

Structure of the semantic interoperability problem

Four levels of organisation of information sharing same semantics:

- The **cognitive user interface** – flexible approach to data capture and viewing
- The **data capture sets** for each step – process-oriented, may be *ad hoc*
- Standardised **semantics of the data points** in data capture sets
- Standardised **data representation**, enabling interoperability
- + Standardised **querying** capability
- + Standardised interface to **terminology** for inferring



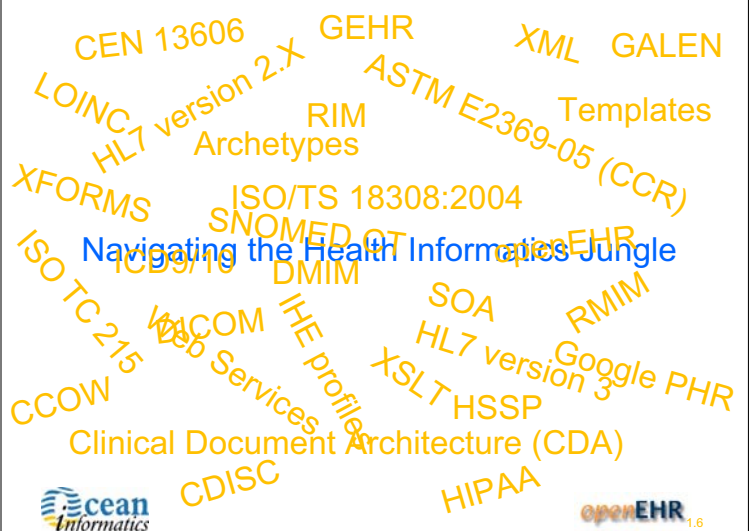
openEHR

In other words....

- It is not just about what is 'on the wire' between two systems....
- A **message-based approach** to semantic interoperability will be largely deficient in the semantics of data capture, definition, re-use and querying.



openEHR



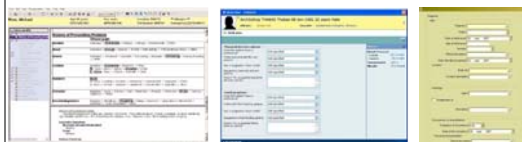
openEHR_{1.6}

openEHR as the Health Computing Platform

- openEHR is **engineered as an EHR architecture** not a messaging solution.
- openEHR is engineered for **semantic interoperability**
- openEHR is engineered to provide a solution for the **logical record architecture**
- openEHR is **engineered to work** and is not just an academic exercise



Information Level 1 & 2



The cognitive User interface:
Different ways of Presenting & Capturing the Same information



Logical data-sets:
Achieved by templates
That re-use and Organise underlying Standardised data
Points *according to use*



Information Level 2 & 3



Logical data sets:
Templates – using only Selected items from a Number of archetypes

Standardised models of The data:
Achieved by archetypes Organised by topic, *Independent of use*



Information Level 3 & 4



Standardised clinical models of the data:
Archetypes – all based On same reference model

Standardised technical representation of the data:
The reference model – Enables interoperability



Queries based on archetypes - Archetype Query Language (AQL)

SELECT

```
o/data[at0001]/events[at0002]/time,
o/data[at0001]/events[at0002]/data[at0003]/items[at0013.1]/value
```

FROM

```
Ehr[uid=@EhrUid] CONTAINS Composition
c[openEHR-EHR-COMPOSITION.encounter.v1]
CONTAINS Observation o[openEHR-EHR-OBSERVATION.laboratory-lipids.v1]
```

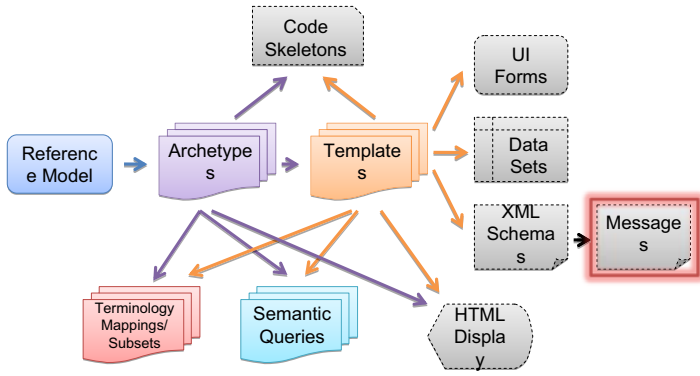


The result...

- **Semantic coherence** in the application stack (all layers of software know what the data mean)
- A **high level of re-use** of artefacts – define once, reuse many times
- A **single, stable reference model** for sharing clinical and related information
- A standardised query language for writing **portable queries**
- A standardised, re-usable way of connecting to **terminology**



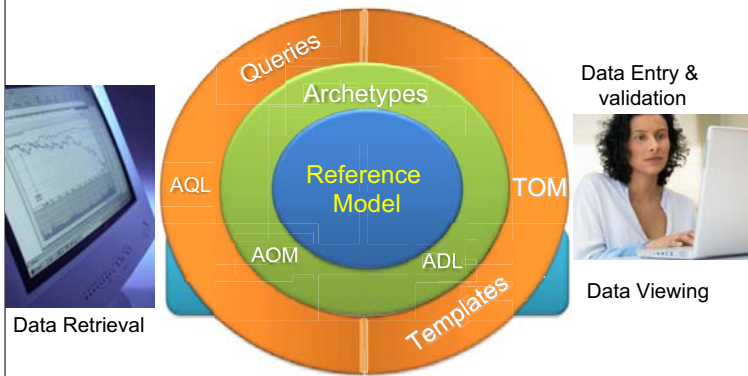
The openEHR artefact ecosystem



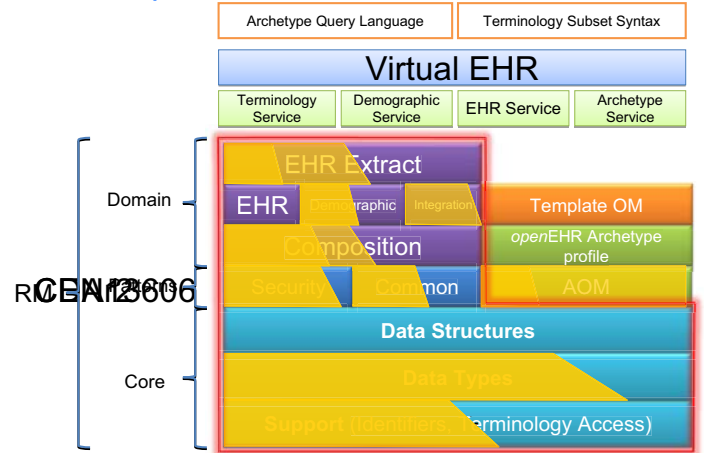
openEHR + Terminology

- Terminologies such as SNOMED CT **don't** solve the interoperability problem
- Archetypes and Templates **work well** with Terminologies
- Terminologies important for capturing concepts that need to be **queried** or shared - subsets
- Place of terminology is for **inferencing** and connecting concepts in an **ontology of reality**

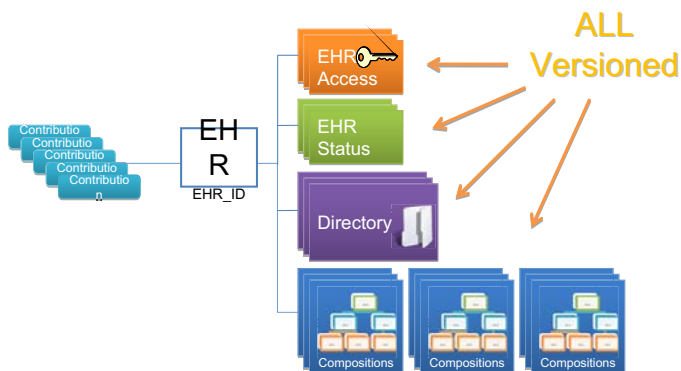
openEHR Health Information Platform



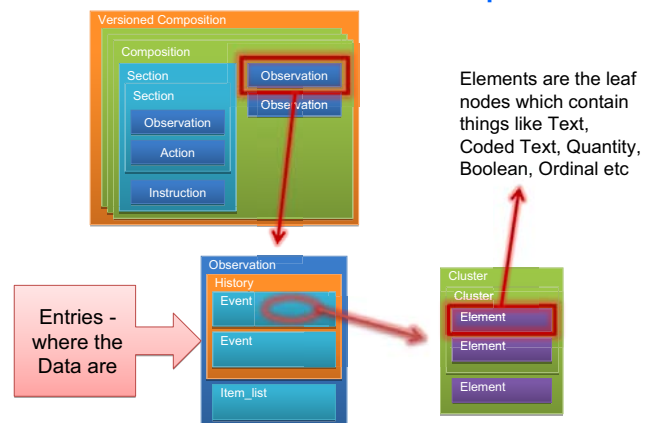
openEHR Reference Model



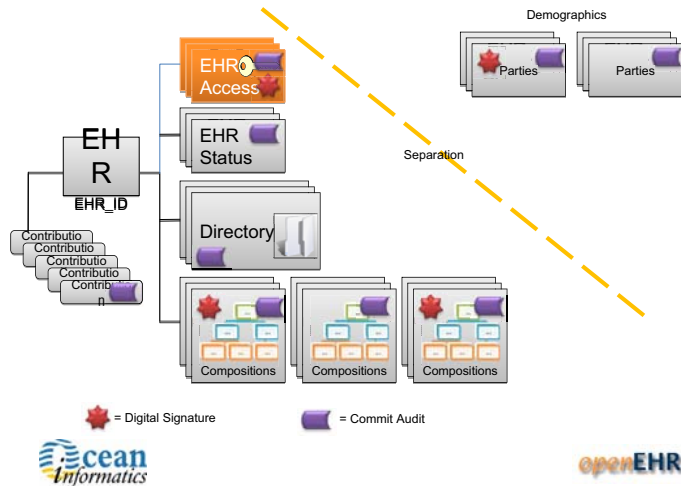
The reference model – structure of one EHR



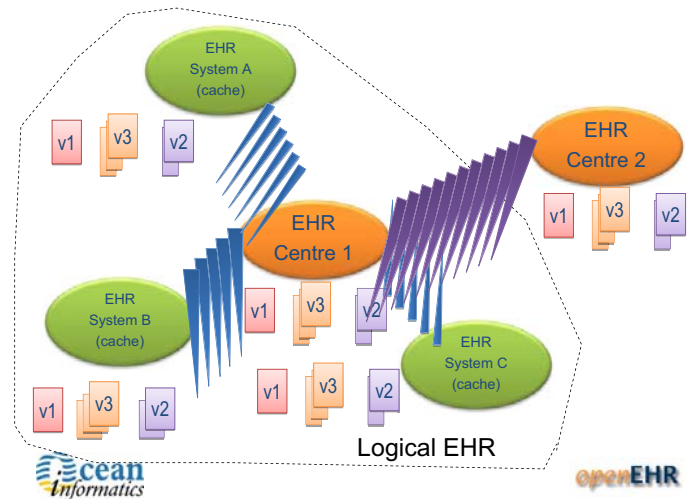
Structure of one Composition



Security Features

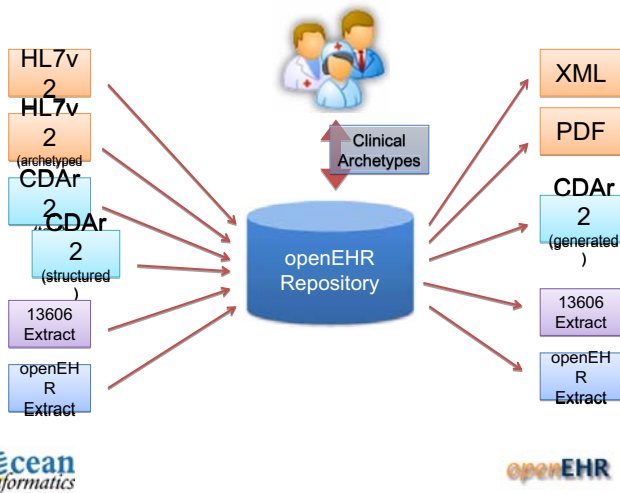


Distributed versioning



A Universal EHR

1.21



Software demonstration:

- Archetype Editor
- Template Designer
- GUI development with templates
- Data persistence and retrieval
- Archetype query language examples
- Terminology subsets