



From semantics to services
– the *openEHR* experience
OMG SOA in HEALTHCARE CONFERENCE
13–15 JULY 2011

Thomas Beale, Washington, 13 July 2011

What are we trying to do
today?

Ultimate ICT goals

- To **Compute** with health information
 - Cross-enterprise
 - Patient-centric
 - Over time & technology changes

Ultimate ICT goals

- In particular:
 - X-enterprise patient care pathway tracking
 - Decision support for doctors
 - Business process analysis for provider orgs
 - Business intelligence for payers & public health
 - Medical research 'study' analysis
 - Person-centred data analysis, ethically targetted marketing etc
 - Integrate health data with social & educational media streams in patient-centred portals

Ultimate ICT goals

- While dealing with relentless change in
 - Medicine, esp. drugs, interactions, procedures...
 - Information
 - Care processes
 - Patient needs
 - Legislation

Getting there requires...

- A ~change-immune **semantic architecture**, allowing
 - **meaning** of information and healthcare process steps etc to be safely and reliably defined
 - **convertability** of information from proprietary / legacy sources to common formats

Getting there requires...

- A systems and **services architecture** defining groupings and access protocols enabling:
 - **Aggregation** of information from source systems
 - Varying **levels of conformance**, esp. for existing systems
 - **Incremental deployment**
 - Satisfying **changing business** needs

Assumptions

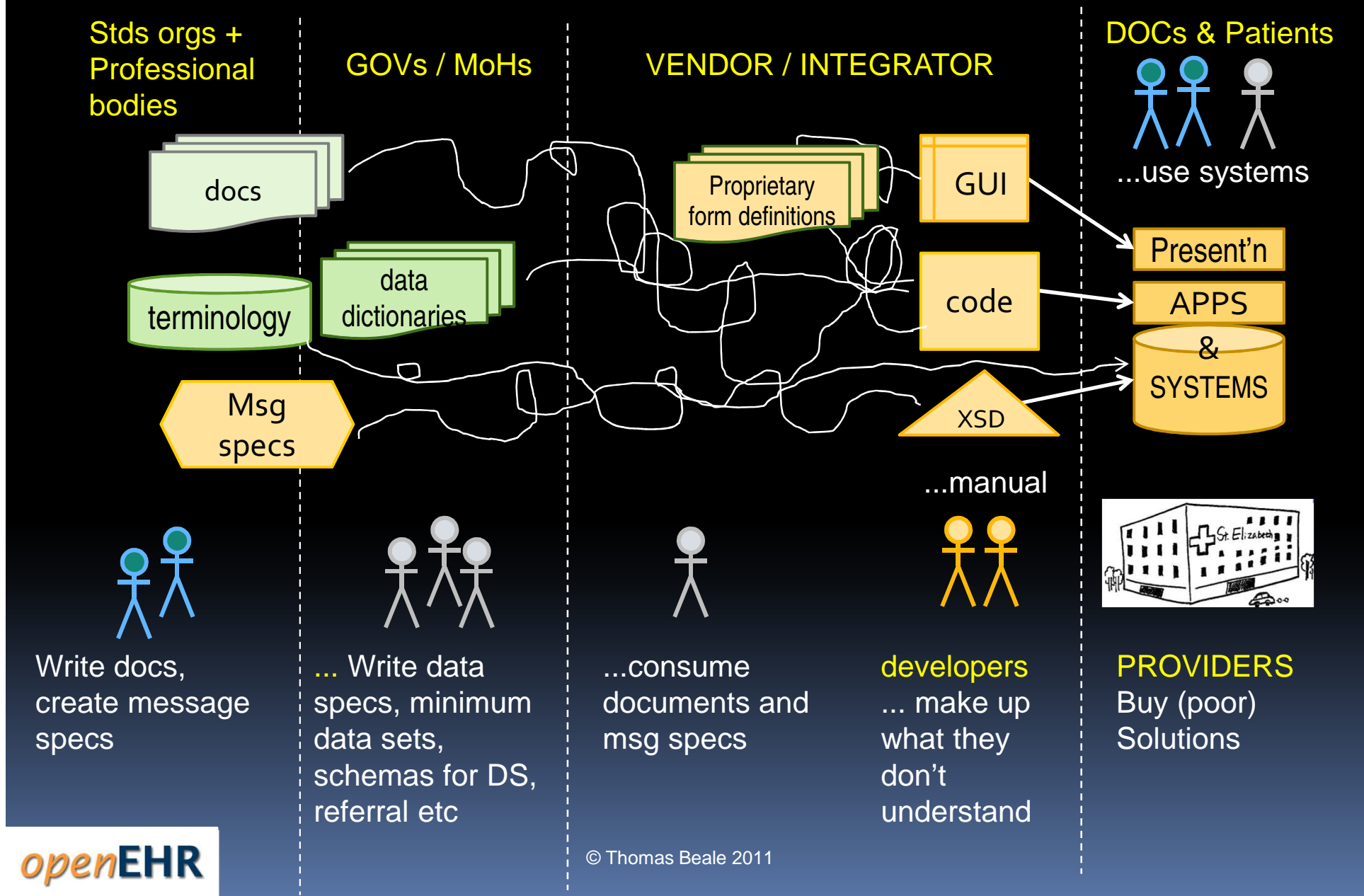
- A services architecture with no semantic underpinning can deliver:
 - Human – human information transmission
 - Very basic search facilities
 - Limited computability, where information is already widely standardised, e.g. HL7v2 lab, ADT
 - Some security / privacy support
- But not generalised patient-centric computability – can't access the main economic potential

The clock is ticking...

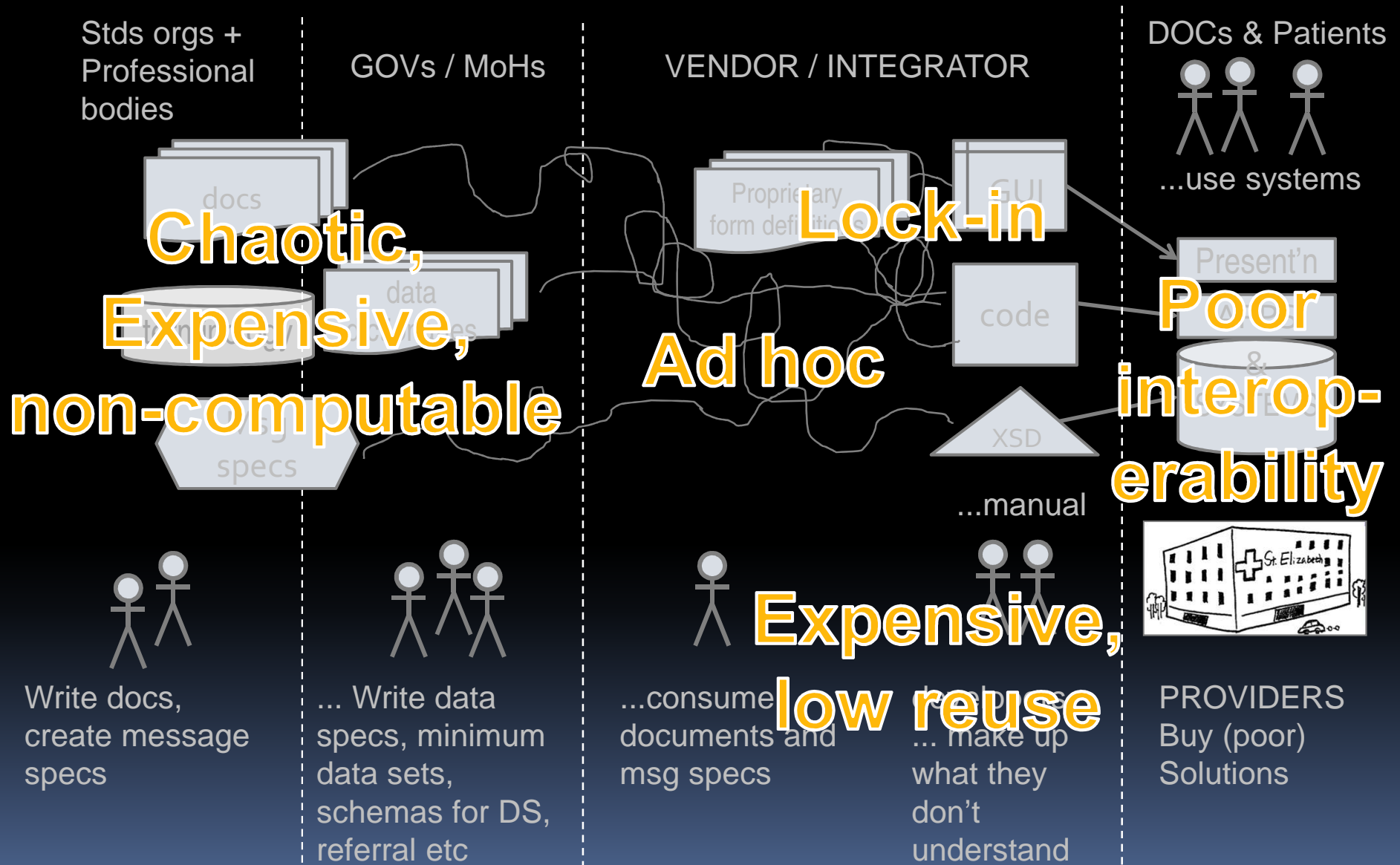
- Today we are still creating peta-bytes of non-interoperable, non-computable health data
- Post-hoc 're-engineering' of the data to make it computable is too expensive to be realistic
- We know this because medical research projects regularly burn their entire budget on data re-engineering

The semantic part

Historical Industry Structure



Historical Industry Structure



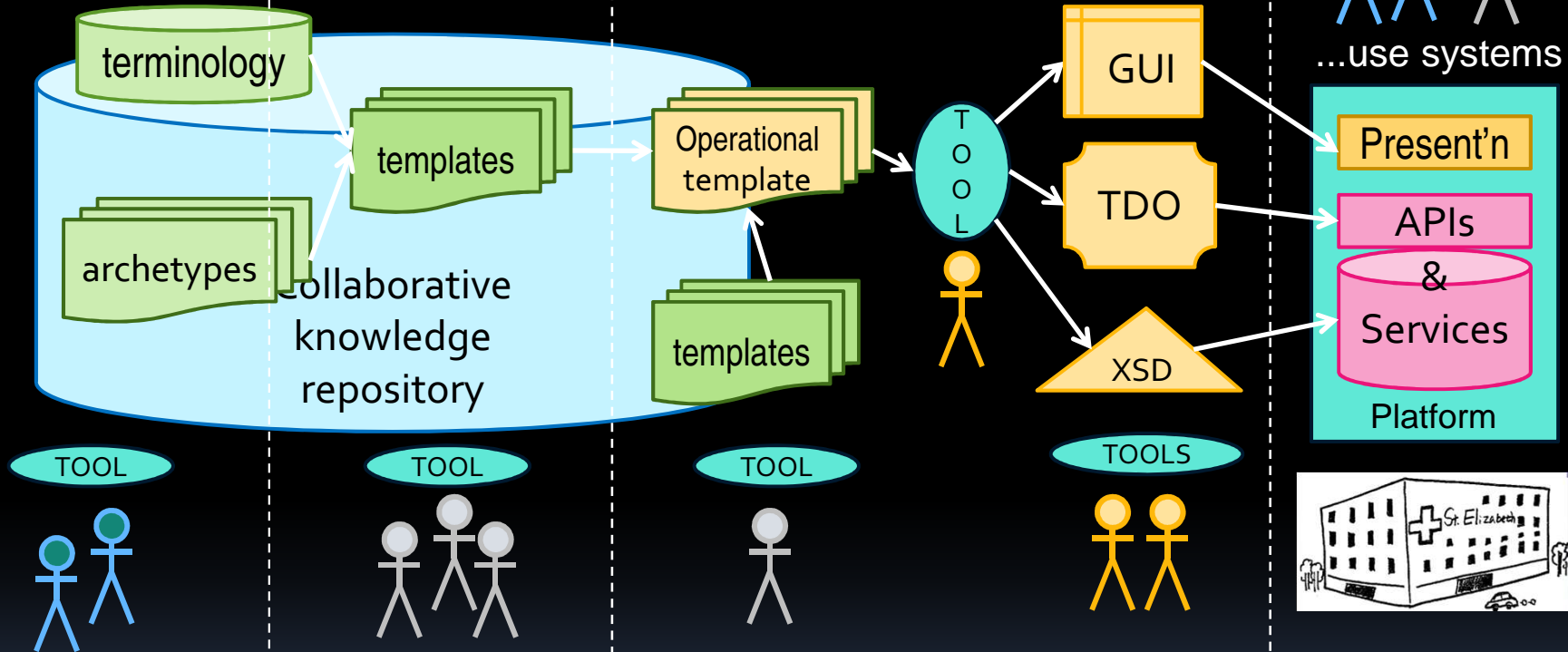
openEHR approach

Std's orgs +
Professional
bodies

GOVs / MoHs

VENDOR / INTEGRATOR

DOCs & Patients



build archetypes
& terminology that
define their
Information –
e.g. via IHTSDO

...build templates
and issue as
standards
e.g. Discharge
Summary

...consume
std templates and
create their own,
making OPTs

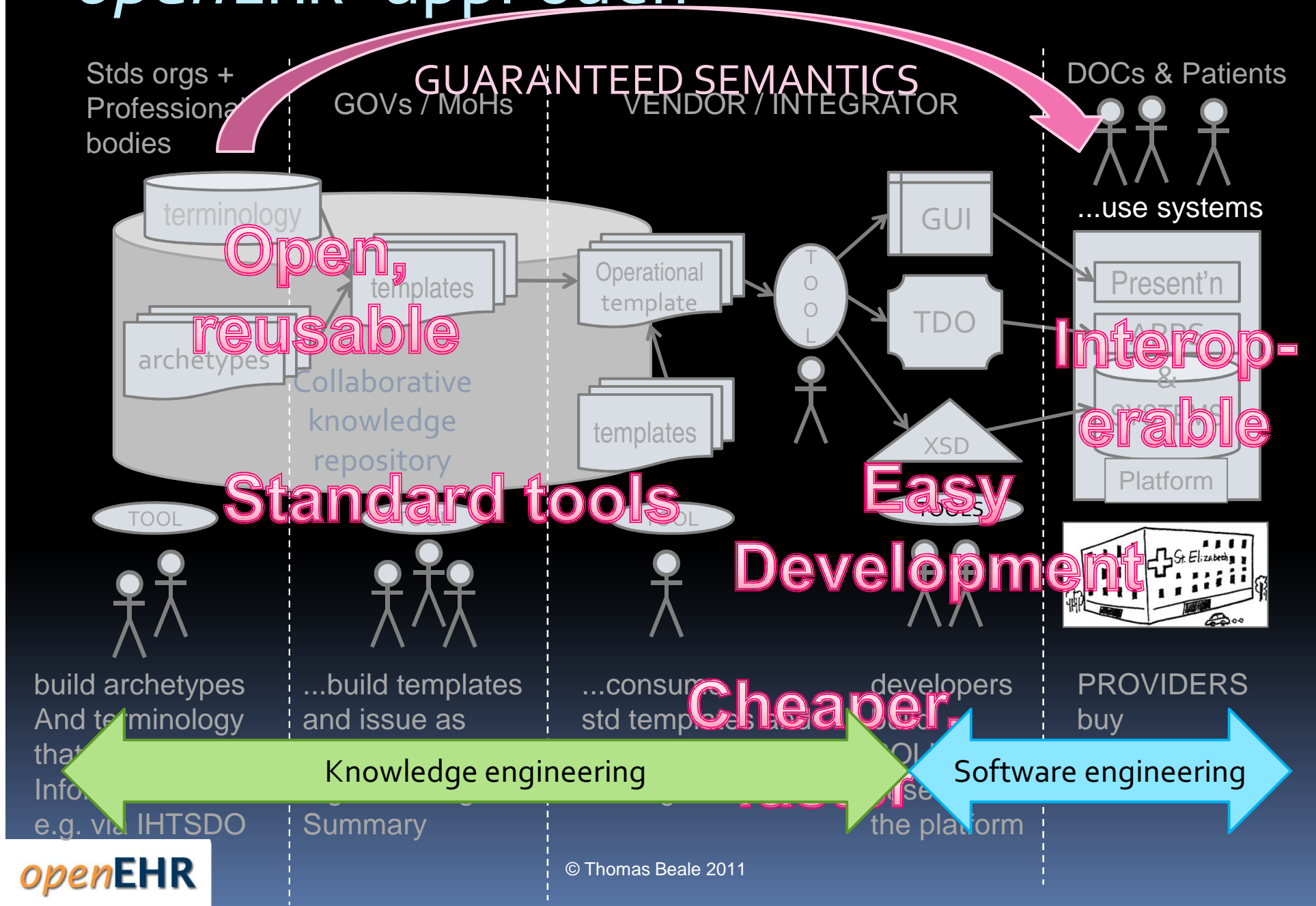
developers
build
SOLUTIONS
based on
the platform

PROVIDERS
buy
Solutions

openEHR

© Thomas Beale 2011

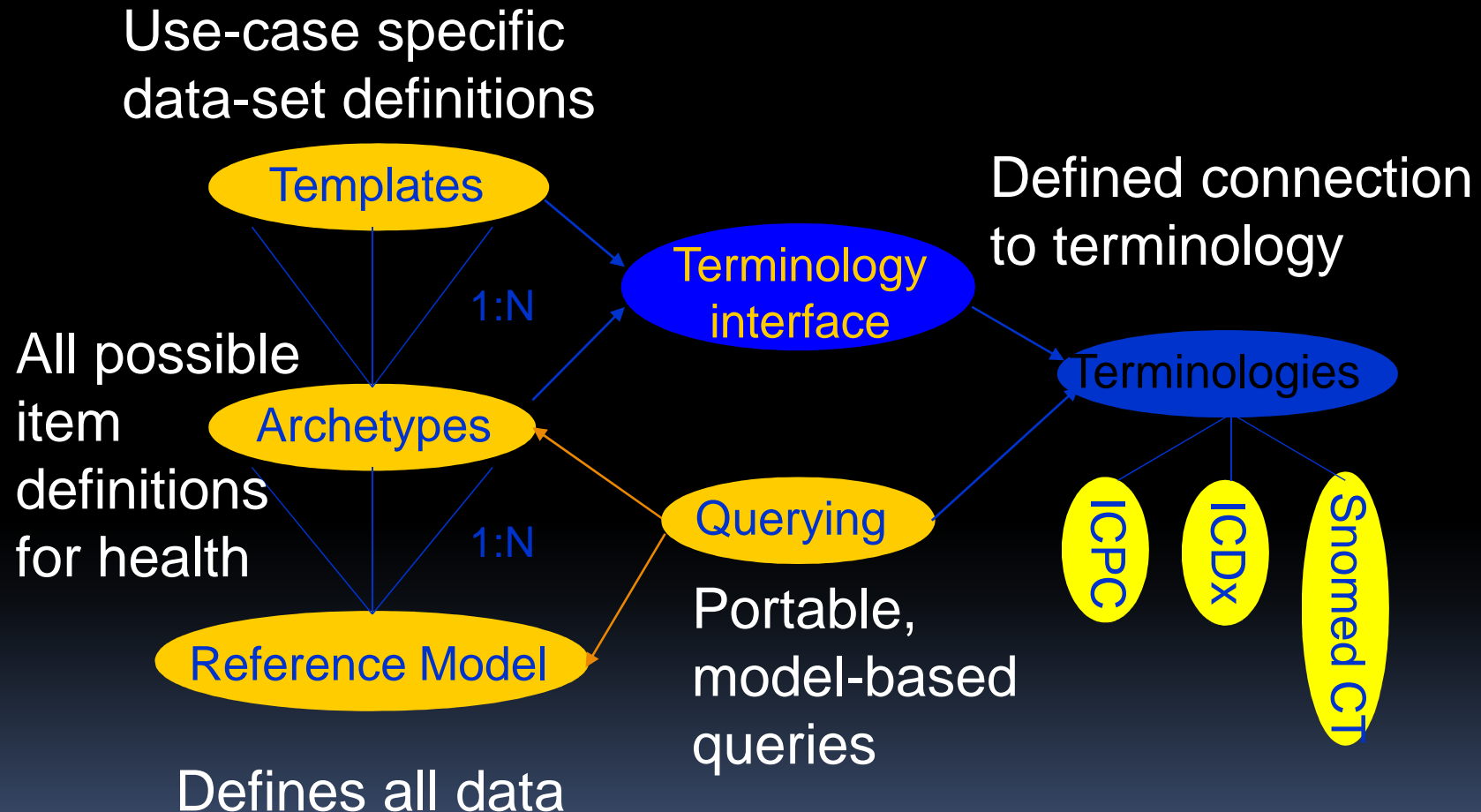
openEHR approach



The basic plan

- A general theoretical paradigm or **framework**
- An **architecture** specific to the domain, including
 - Actual specifications for formalisms, models etc
- Actual **models**

The *openEHR* framework

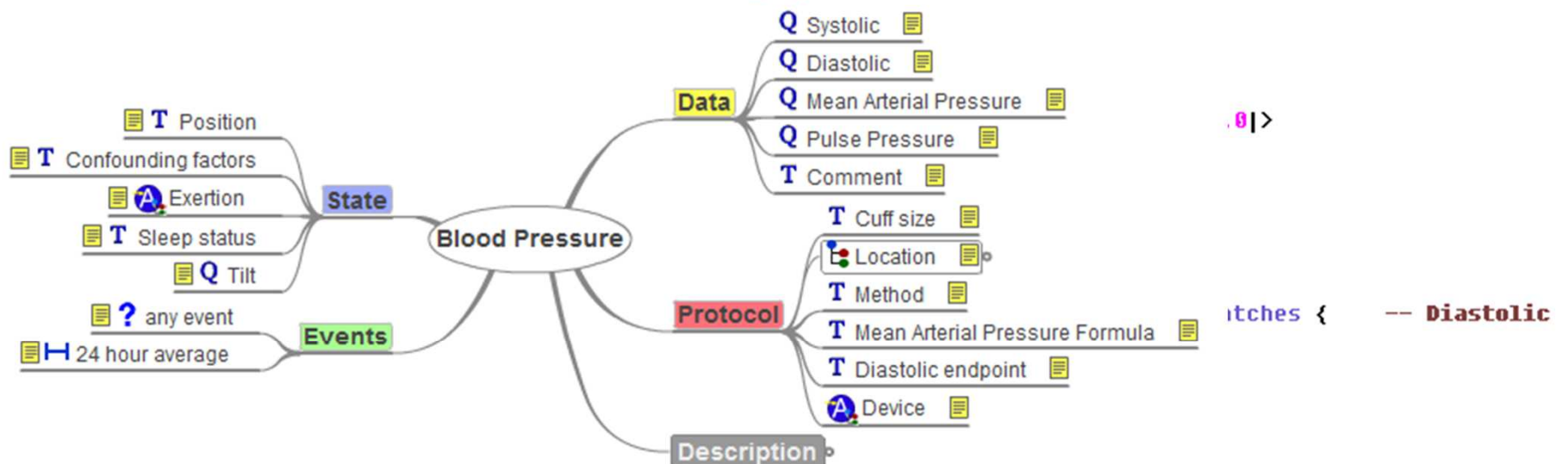


openEHR Archetype

```

definition
  OBSERVATION[at0000] matches { -- Blood Pressure
    data matches {
      HISTORY[at0001] matches { -- history
        events cardinality matches {1..*; unordered} matches {
          EVENT[at0006] occurrences matches {0..*} matches { -- any event
            data matches {
              ITEM_TREE[at0003] matches { -- blood pressure
                items matches {
                  ELEMENT[at0004] occurrences matches {0..1} matches { -- Systolic
                    value matches {
                      (C_DV_QUANTITY) <

```



```

magnitude = <|0.0..1000.0|>
precision = <|0|>

```

AQL query

- SELECT com2/context/start_time/value as START_DATE,
obs1/data[at0001]/events[at0006]/data[at0003]/items[at0004]/value/magnitude as
SYSTOLIC,
obs1/data[at0001]/events[at0006]/data[at0003]/items[at0005]/value/magnitude as
DIASTOLIC,
obs3/data[at0002]/events[at0003]/data[at0001]/items[at0004]/value/magnitude as
PULSE_RATE,
obs4/data[at0001]/events[at0002]/data[at0003]/items[at0004]/value/magnitude as
RESPIRATORY_RATE....
- FROM EHR e[ehr_id/value='f271cd26-23fc-43a1-b411-34cdadaeao67'] CONTAINS
COMPOSITION com2 [openEHR-EHR-COMPOSITION.encounter.v1]
- CONTAINS (OBSERVATION obs3 [openEHR-EHR-OBSERVATION.heart_rate-pulse-
zn.v1] OR OBSERVATION obs1 [openEHR-EHR-OBSERVATION.blood_pressure-zn.v1]
OR OBSERVATION obs4 [openEHR-EHR-OBSERVATION.respiration.v1]
- WHERE com2/name/value matches {'Vital functions', 'Respiratory assessment',
'Assessment scales'} AND obs9/name/value = 'PEF before' AND obs10/name/value =
'PEF after' AND com2/context/start_time >= '20110406T000000.000+0200' AND
com2/context/start_time < '30000101T000000.000+0100'

The diagram illustrates the Software Core Architecture, showing the flow from Generated software to Developed software. The architecture is organized into three main horizontal sections: Generated software (top, light blue), Developed software (top, dark blue), and Software core (bottom, light blue).

Generated software (top left) contains:

- Templates** (yellow oval)
- Archetypes** (yellow oval)
- Reference Model** (yellow oval)

Developed software (top right) contains:

- Terminology interface** (blue oval)
- Terminologies** (blue oval)
- ICPC** (yellow oval)
- ICDx** (yellow oval)
- Snomed CT** (yellow oval)

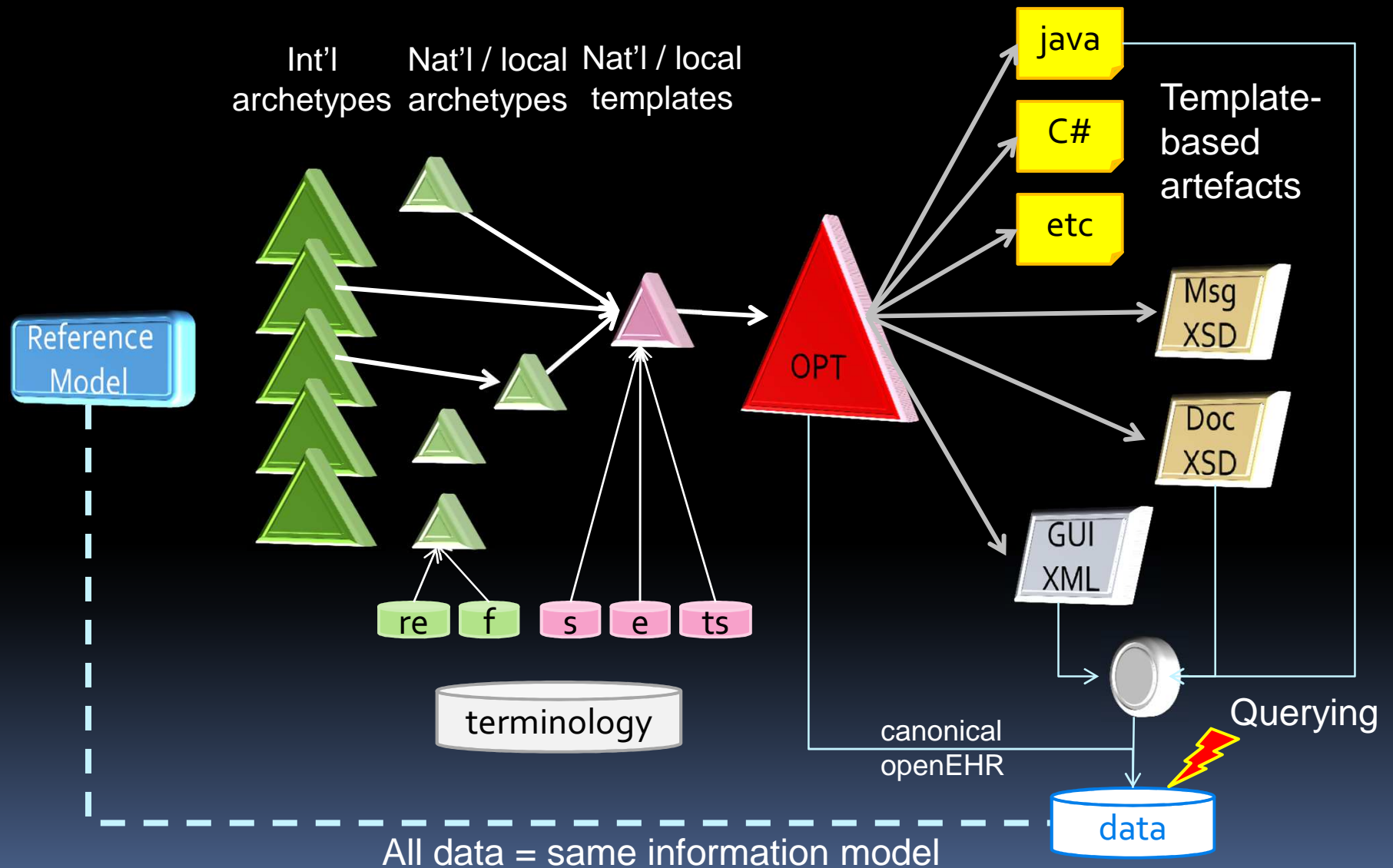
Software core (bottom) contains:

- Querying** (yellow oval)

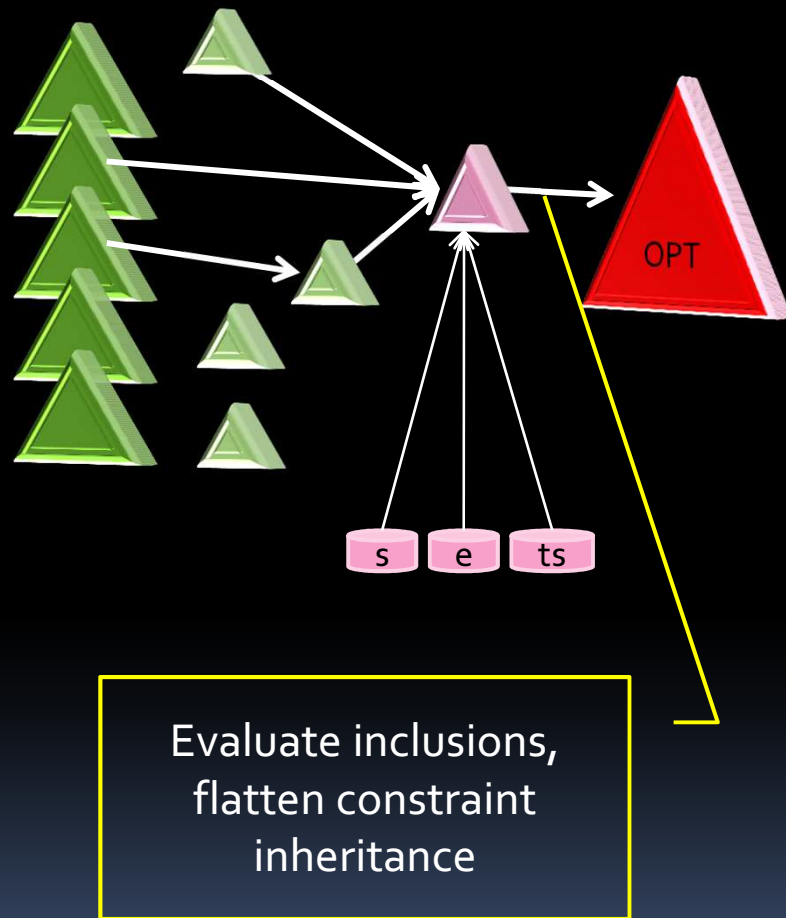
Flow and Relationships:

- Generate:** A yellow arrow points from the **Software core** up to the **Generated software** section.
- Consumption:** Three red arrows point from the **Generated software** section up to the **Developed software** section, labeled **Consume**.
- Internal Flow:**
 - Thin blue arrows connect **Templates** to **Archetypes** and **Archetypes** to **Reference Model**.
 - Thin blue arrows connect **Archetypes** to **Terminology interface** and **Reference Model** to **Querying**.
 - Thin blue arrows connect **Terminology interface** to **Terminologies** and **Querying** to **Terminologies**.
 - Thin blue arrows connect **Terminologies** to **ICPC**, **ICDx**, and **Snomed CT**.
- Relationships:**
 - A **1:N** relationship is indicated between **Templates** and **Archetypes**.
 - A **1:N** relationship is indicated between **Archetypes** and **Reference Model**.

The architecture



The key...

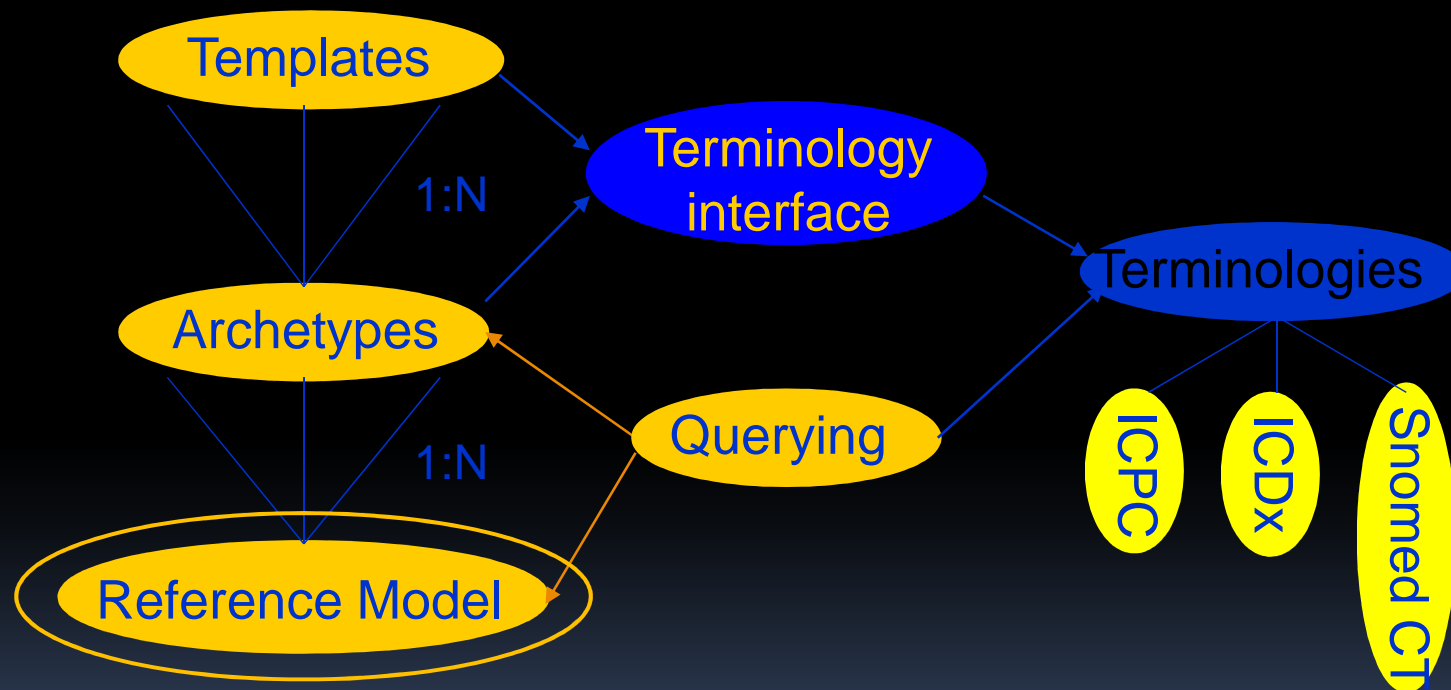


- Is the operational template (OPT) – this is the joining point between the semantic specifications and deployable software artefacts that can be used by normal developers

Key Outcomes

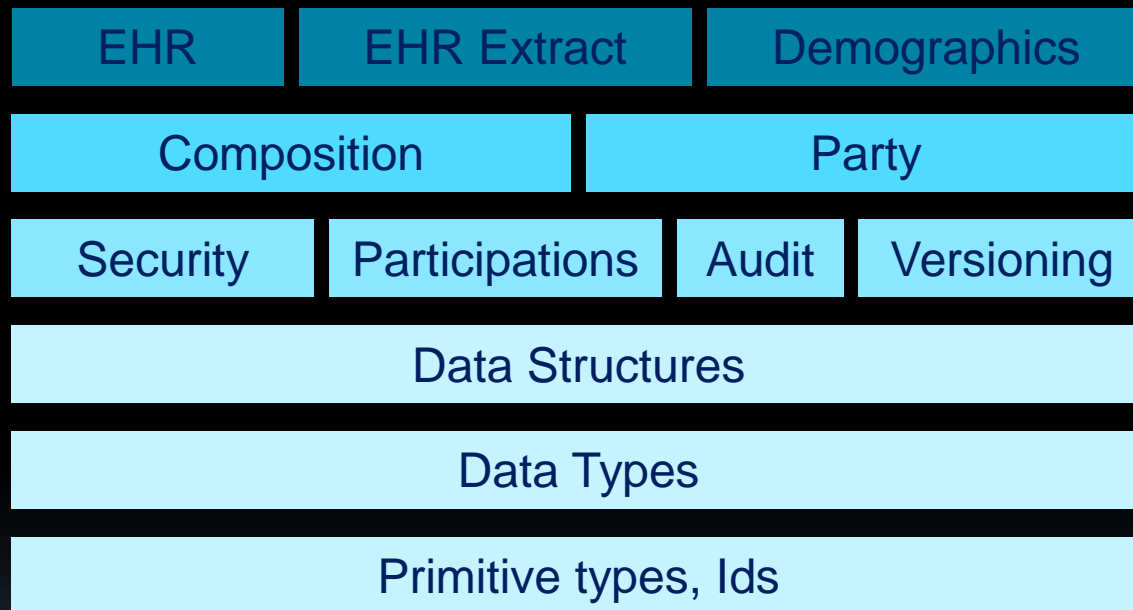
- Normal developers can engage – *openEHR* + Snomed become economic and ~quick
- Semantic connection exists between definitions and implementations
 - → now we know what the meaning of data are, and DS and BI can work...

The *openEHR* framework



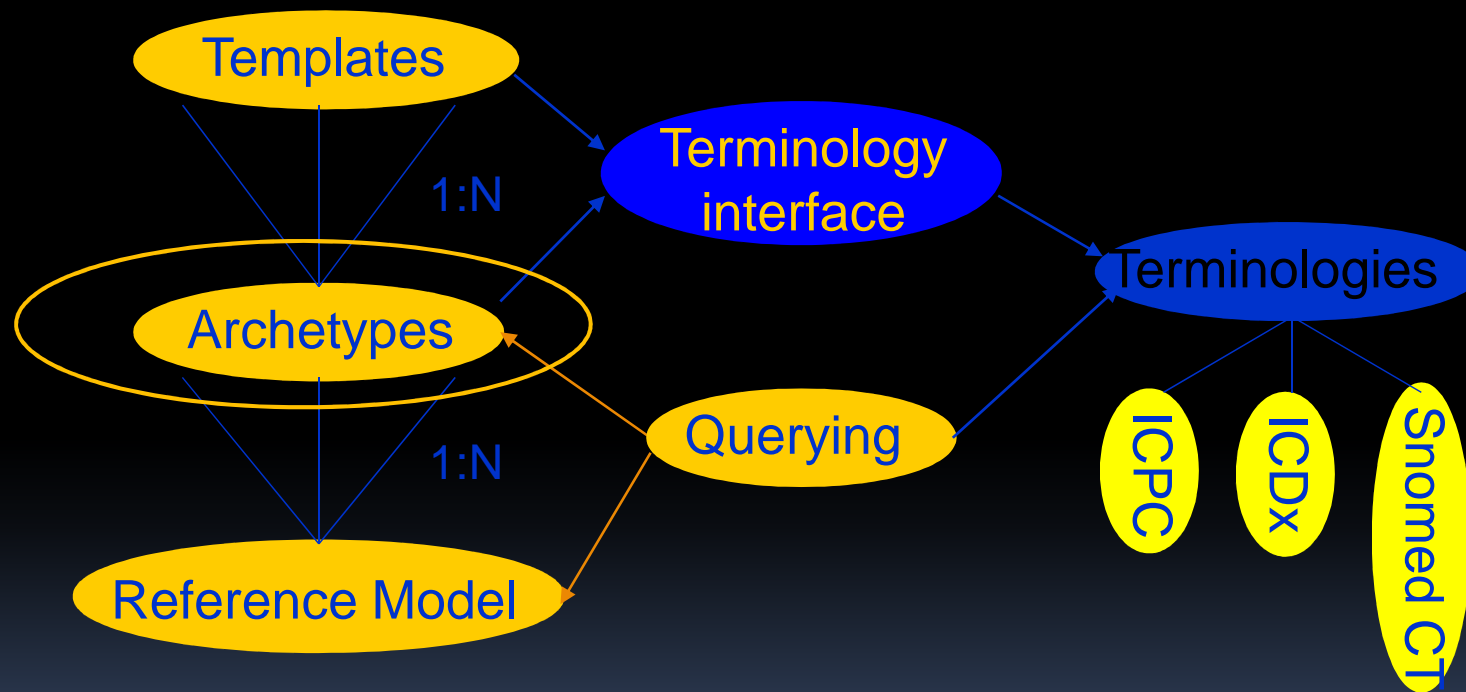
The reference model

Will continue to grow, to accommodate process, workflow etc

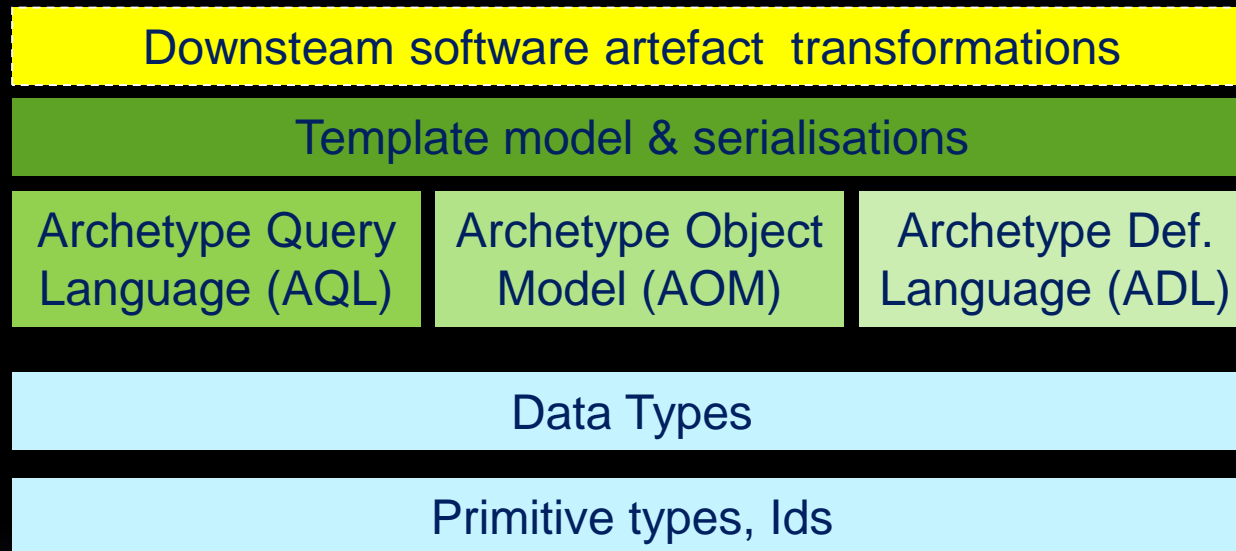


<http://www.openehr.org/releases/1.0.2/roadmap.html>

The *open*EHR framework



The archetype architecture



<http://www.openehr.org/releases/1.0.2/roadmap.html>

Managing knowledge artefacts

- Content models & terminology ref sets managed outside of software process & people
- Needs:
 - Governance
 - Methodology
 - Identification
 - Sharing and release rules
 - etc

Clinical Knowledge Manager

- A tool for involving clinicians in defining clinical content based on archetypes, templates, and termsets

The screenshot displays the openEHR Clinical Knowledge Manager web application. The interface is divided into several sections:

- Header:** Includes the openEHR logo, the title "Clinical Knowledge Manager", and navigation tabs for "Archetypes", "Reviews", "Teams", "Release Sets", "Reports", "Users", and "About". A user greeting "Welcome, Sebastian Garde" and a "Sign out" link are visible on the right.
- Left Sidebar:** A tree view under "Archetypes" showing a hierarchy of "All archetypes" and "Teams: All teams". The "EHR Archetypes" section is expanded, showing various clinical concepts like "Appar score", "Audiogram result", "Autopsy examination", "Barthel index", "Blood Pressure", "Blood gas assessment", "Blood matching", "Body mass index", "Body temperature", "Body weight", "Cancer observation", "Diagnostic imaging", "Distraction Hearing Test", "ECG recording - 12-lead stan", "Electroacoustic Hearing Test", "Examination findings", "Faeces", "Feeding", "Fundoscopic examination of", "Glasgow Coma Scale", "Global assessment", "Heart rate", "Height/Length", "Intravascular pressure", "Laboratory test result", "Measurement of chest and ex", "Menstruation", "Movement of the fetus", "Operation Record", "Oral fluid intake", "Oxymetry", "Postural oedema", and "Pregnancy test".
- Main Content Area:**
 - Find archetypes:** A search bar with "blood" entered. Below it are radio buttons for "Restrict search to main data" (selected) and "Complete search". An "Advanced" link is also present.
 - Search using...:** Radio buttons for "Or" (selected) and "And".
 - Subclasses...:** Radio buttons for "Restrict search to directly selected classes" (selected) and "Include subclasses in search".
 - Find Archetypes:** A button to execute the search.
 - Search results:** A message states "Search results for 'blood' within the main elements of archetypes on the trunk that are active." Below this, it says "Found 12 archetypes." A list of results is shown, each with a magnifying glass icon, a title, and a "Details" link:
 - Blood matching:** Archetype ID: openEHR-EHR-OBSERVATION.blood_match.v1, Status: Draft.
 - Blood gas assessment:** Archetype ID: openEHR-EHR-OBSERVATION.blood_gases.v1, Status: Draft.
 - Blood Pressure:** Archetype ID: openEHR-EHR-OBSERVATION.blood_pressure.v1, Status: Published.
 - Body mass index:** Archetype ID: openEHR-EHR-OBSERVATION.body_mass_index.v1, Status: Published. Location: Value: The Evidence Report [Internet]. Bethesda (MD): National Heart, Lung, and Blood Institute; NIH Publication No. 98-4083, Sep 1998, [cited 2009].
 - Intravascular pressure:** Archetype ID: openEHR-EHR-OBSERVATION.intravascular_pressure.v1, Status: Draft. Location: Misuse: Not to be used for systemic blood pressure. Use observation.blood_pressure for this.

CKM Core Principles

Separation of technical and clinical aspects to successfully involve clinicians in

- Informal Discussions
- Formal Reviewing (content, terminology binding, translations)
- Sharing
- Publishing
- Revision/Version Management
- Release and Dependency Management

CKM Approach

- Web 2.0 design
 - Easier to engage clinicians: Can now use 5 mins or 1 hour of an expensive specialised clinician's time; before, they lost hours on physical meetings
- Implementation is growing as we learn
 - Can respond quickly to changing needs, evolving methodology
- More than a tool
 - Engage and manage the community

CKM Users

- International *openEHR* CKM instance
 - > 630 users
 - From 64 countries
- National programs with an instance of CKM
 - Australia: Nehta
 - Sweden: SKL

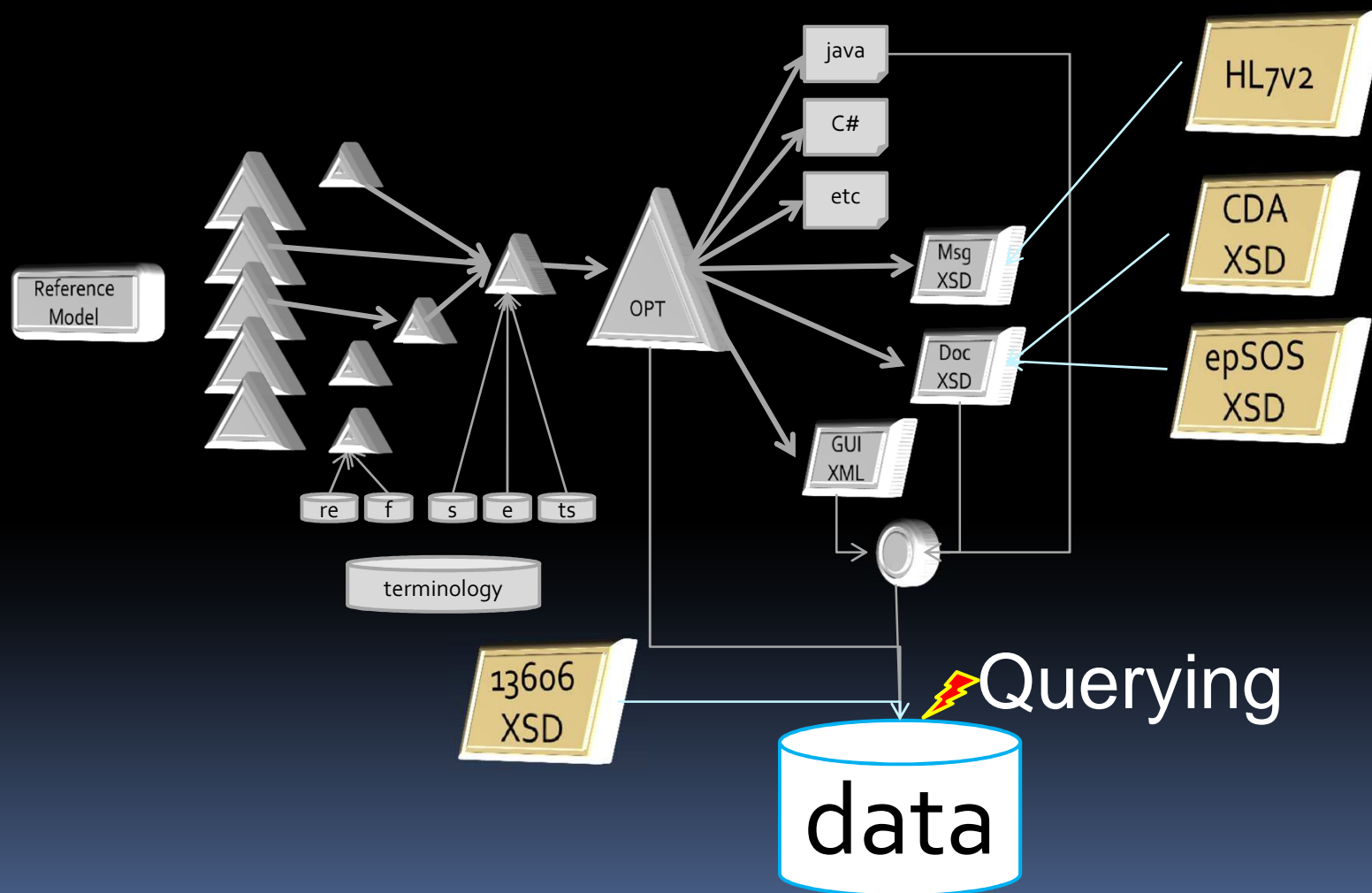
Key Messages

- Knowledge Management is crucial
 - High-quality archetypes with high-quality clinical content
 - Semantically interchangeable between clinical systems; also the basis for decision support
 - Key to success: how to engage with clinicians and capture their knowledge
- CKM - <http://www.openehr.org/knowledge>

Key Outcomes

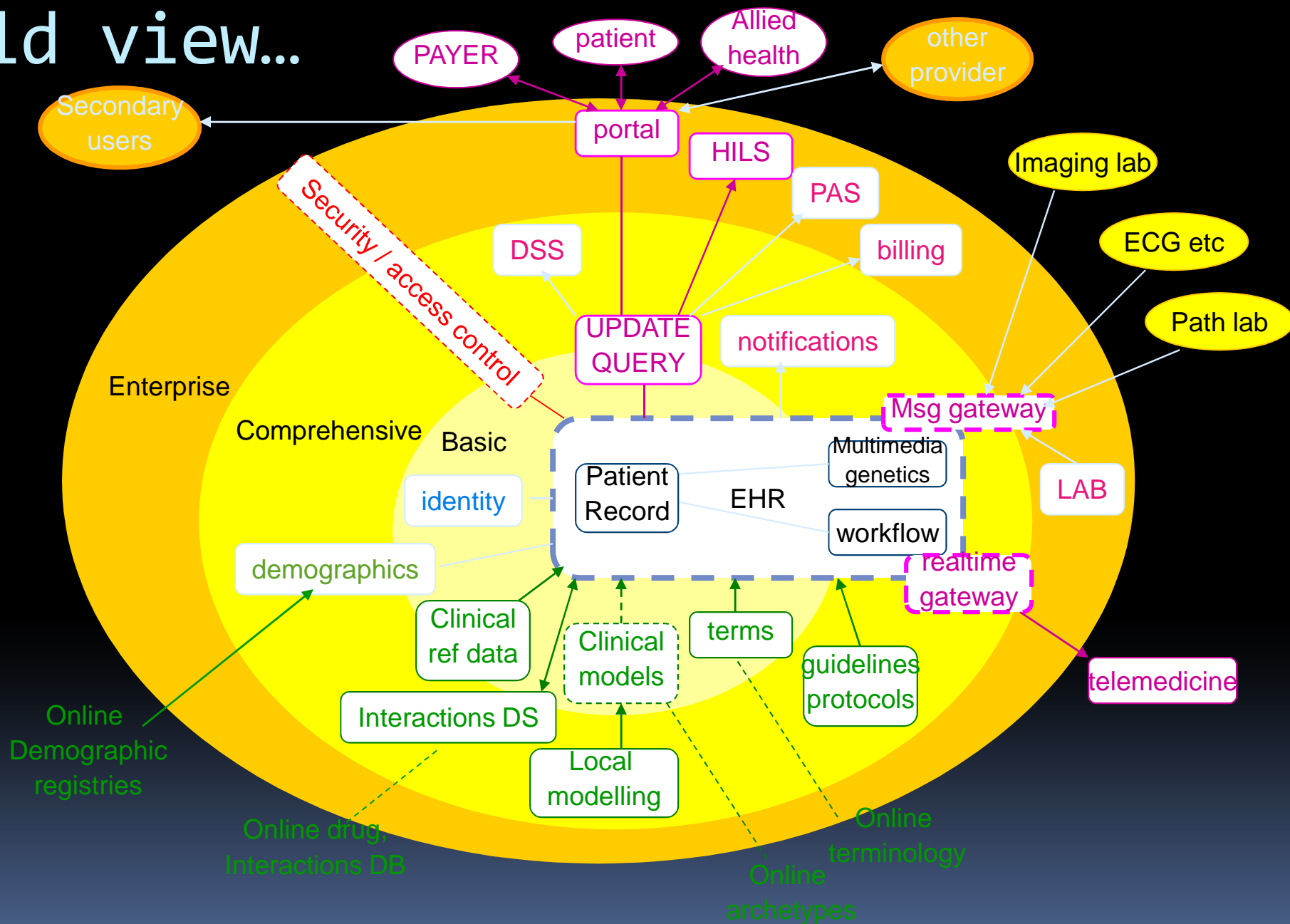
- We can now start to situate existing standards in the framework
 - Concrete content-specific standards like HL7 message definitions, CDAs, CCRs etc are DOWNSTREAM generations and/or mappings of operational templates
 - Meaning we can connect them into a semantic framework and potentially guarantee their semantics
 - Rather than manually building them in a standalone fashion

Tool-based standards



The Services Part

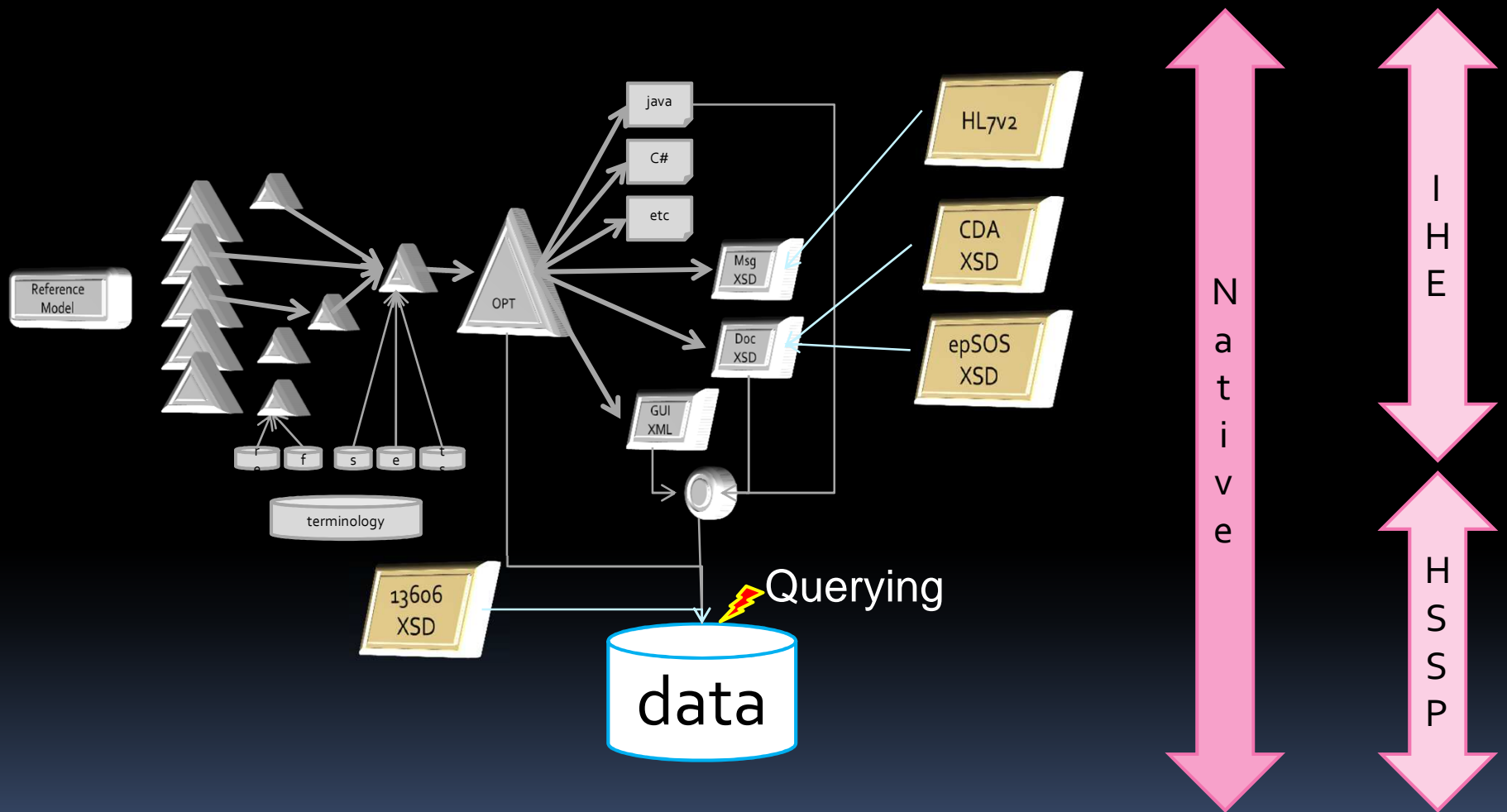
Old view...



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

Services

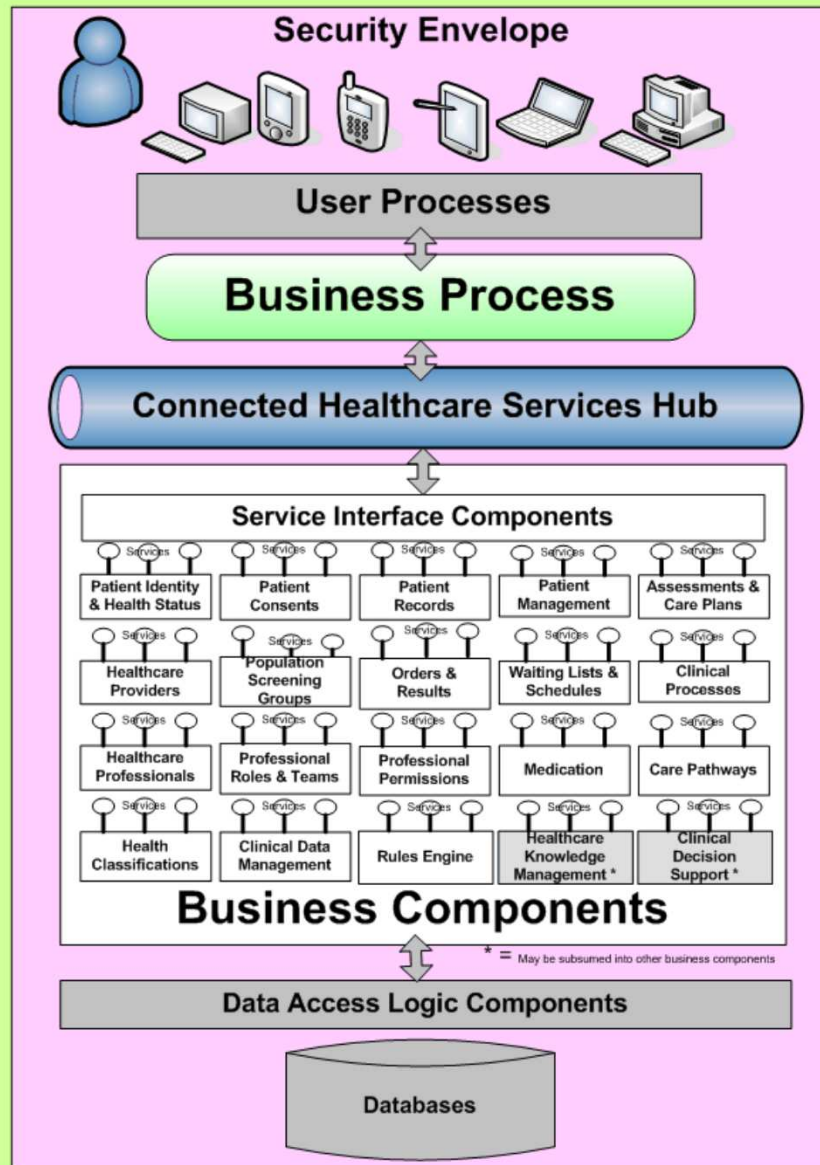


On Services...

- 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

Communications

Operations Management

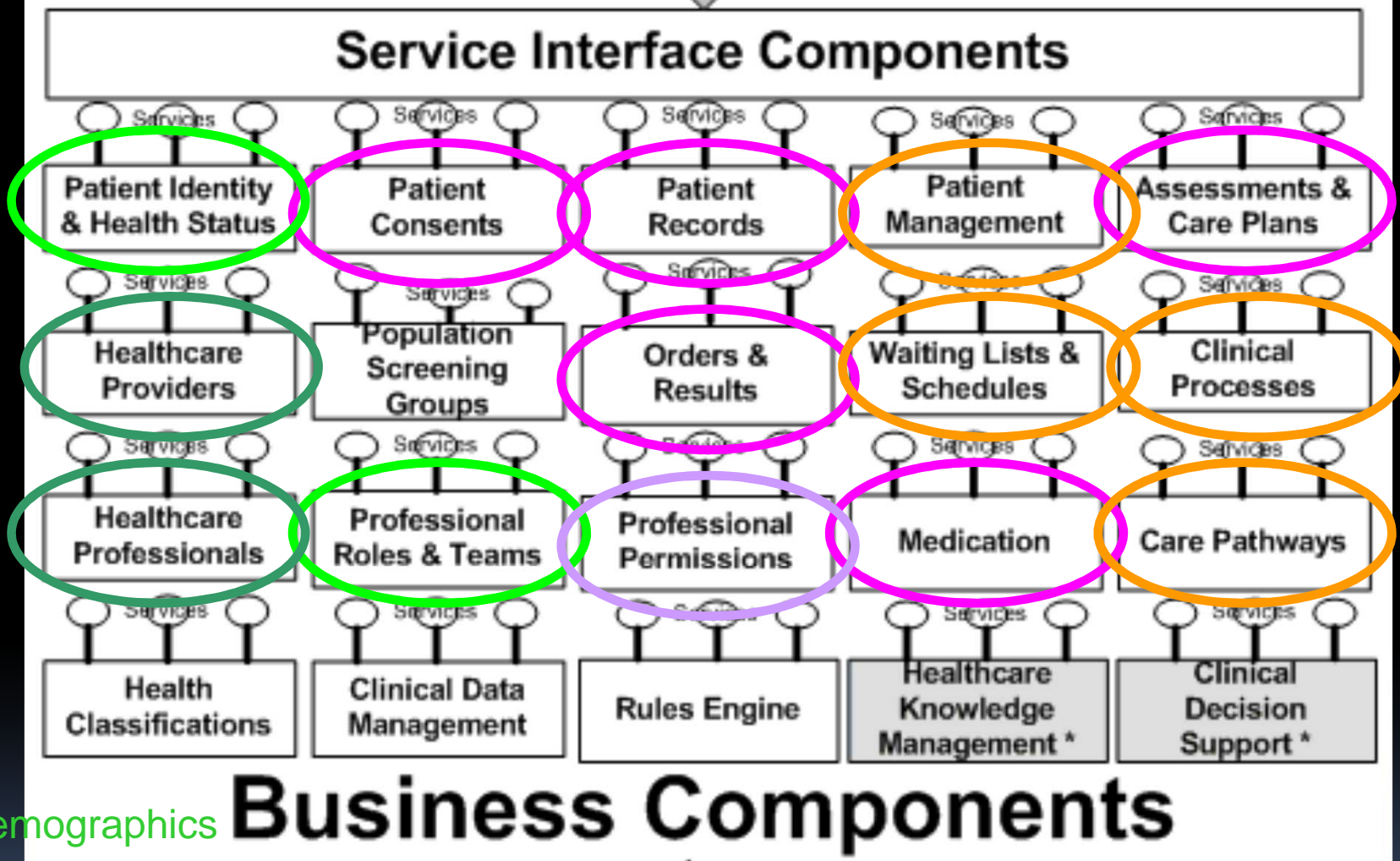


Microsoft CHF

- ProcessModels
- Service Models
- Information Models

EHR Service

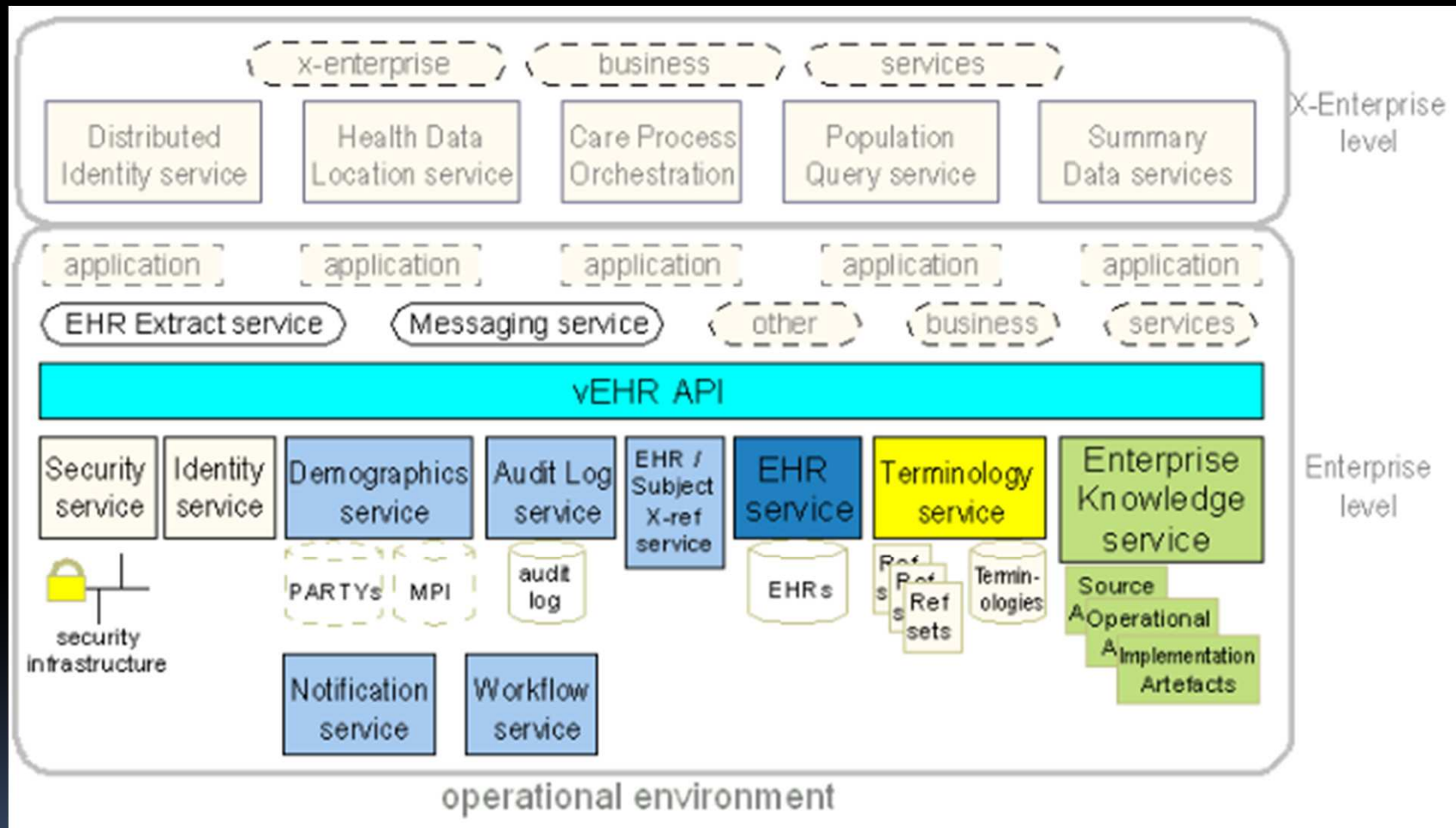
Workflow & Pathway Services



demographics

Security & privacy

The openEHR services architecture



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

Native services

- All services archetype/template aware
- 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

Key elements that MUST WORK

- Everything in openEHR relies on **archetype paths**, which are X-path compatible
- The two tests are being able to:
 - Write portable queries
 - Create URIs to finest grain of data

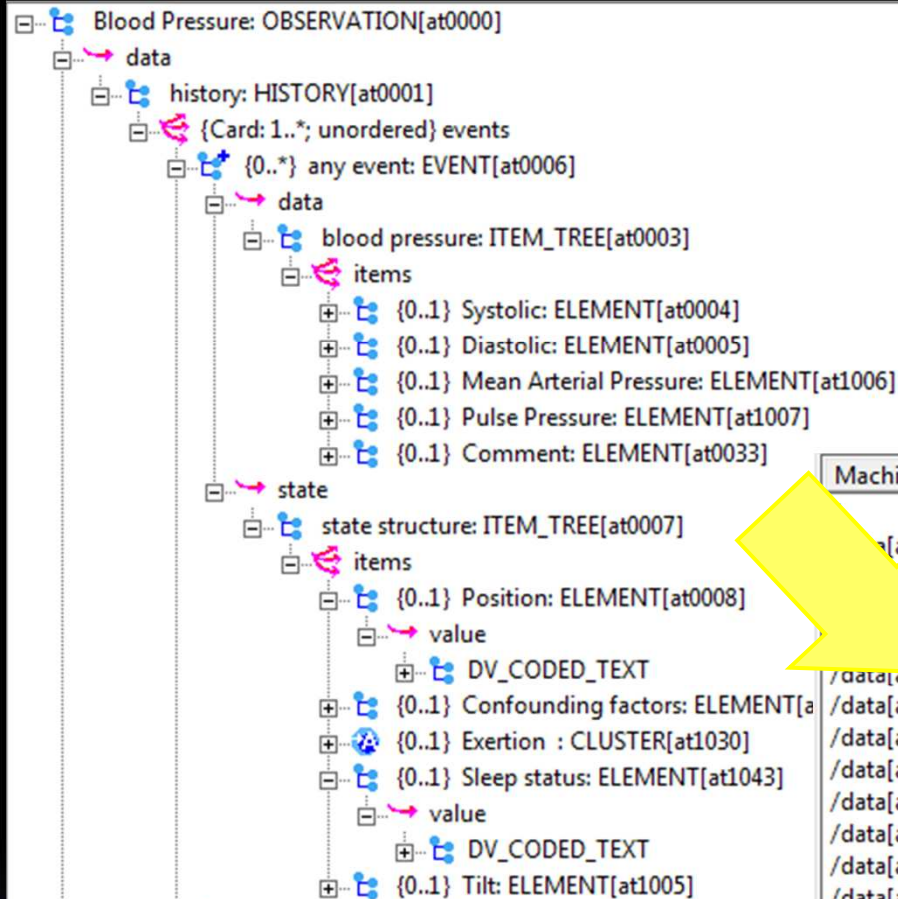
AQL query

- SELECT com2/context/start_time/value as START_DATE,
obs1/data[at0001]/events[at0006]/data[at0003]/items[at0004]/value/magnitude as
SYSTOLIC,
obs1/data[at0001]/events[at0006]/data[at0003]/items[at0005]/value/magnitude as
DIASTOLIC,
obs3/data[at0002]/events[at0003]/data[at0001]/items[at0004]/value/magnitude as
PULSE_RATE,
obs4/data[at0001]/events[at0002]/data[at0003]/items[at0004]/value/magnitude as
RESPIRATORY_RATE....
- FROM EHR e[ehr_id/value='f271cd26-23fc-43a1-b411-34cdadaeao67'] CONTAINS
COMPOSITION com2 [openEHR-EHR-COMPOSITION.encounter.v1]
- CONTAINS (OBSERVATION obs3 [openEHR-EHR-OBSERVATION.heart_rate-pulse-
zn.v1] OR OBSERVATION obs1 [openEHR-EHR-OBSERVATION.blood_pressure-zn.v1]
OR OBSERVATION obs4 [openEHR-EHR-OBSERVATION.respiration.v1]
- WHERE com2/name/value matches {'Vital functions', 'Respiratory assessment',
'Assessment scales'} AND obs9/name/value = 'PEF before' AND obs10/name/value =
'PEF after' AND com2/context/start_time >= '20110406T000000.000+0200' AND
com2/context/start_time < '30000101T000000.000+0100'

openEHR URI

- ehr:1234567/87284370-2D4B-4e3d-A3F3-F303D2F4F34B@latest_trunk_version/content[openEHR-EHR-SECTION.vital_signs.v1]/items[openEHR-EHR-OBSERVATION.heart_rate-pulse.v1]/data/events[at0006, 'any event']/data/items[at0004]

Where paths come from



Machine	RM Type
/data[at0000]	OBSERVATION
/data[at0000]/history[at0001]	HISTORY
/data[at0000]/history[at0001]/events[at0006]	EVENT
/data[at0000]/history[at0001]/events[at0006]/data[at0003]	ITEM_TREE
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at0004]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at0004]/value	DV_QUANTITY
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at0005]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at0005]/value	DV_QUANTITY
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at0033]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at0033]/value	DV_TEXT
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at1006]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at1006]/value	DV_QUANTITY
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at1007]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/data[at0003]/items[at1007]/value	DV_QUANTITY
/data[at0000]/history[at0001]/events[at0006]/state[at0007]	ITEM_TREE
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at0008]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at0008]/value	DV_CODED_TEXT
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at0008]/value/defining_code	CODE_PHRASE
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at1005]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at1005]/value	DV_QUANTITY
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at1030]	CLUSTER
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at1043]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at1043]/value	DV_CODED_TEXT
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at1043]/value/defining_code	CODE_PHRASE
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at1052]	ELEMENT
/data[at0000]/history[at0001]/events[at0006]/state[at0007]/items[at1052]/value	DV_TEXT
/data[at0000]/history[at0001]/events[at1042]	INTERVAL_EVENT

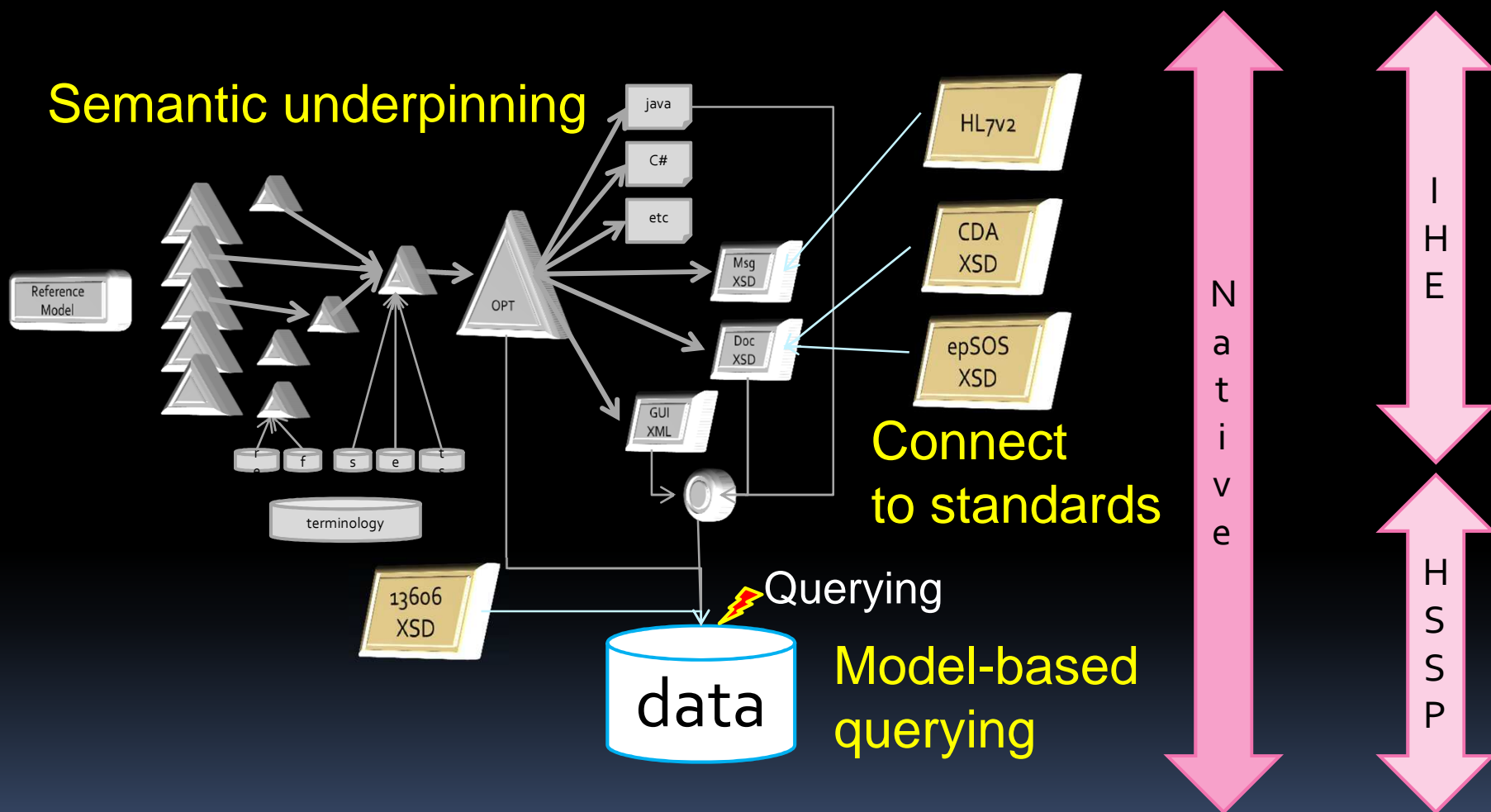
Conclusions

Basic premise

- If we want to share and compute with health data at any level of detail, we need a knowledge-based architecture

Architecture

Knowledge-enabled
services



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 DCMs

The ultimate test

- If you can create a URI to 'my instantaneous resting, lying down systolic BP, recorded 7/jan/2011' then you can communicate at any level of detail about health information