ORACLE®

Health Care Support Beyond Record Keeping

Dieter Gawlick

Agenda

- Patient care
  - Exiting IT support for patient care
  - IT requirement for patient care
  - Extracting evidence from patient data
- The technology
  - Continuous queries
  - Issues with CQ technology
  - Registered queries – state based event processing
  - Information and knowledge discovery
- Conclusion

Introduction

- 4% of US physicians report having an extensive, fully functional electronic records system
- 13% of US physicians report using a basic electronic system
- 1.5% of US hospitals have a comprehensive electronic records system
- 7.6% of US hospitals have a basic system

DesRoches CM et al. Electronic health records in ambulatory care – a national survey of physicians. N Engl J Med 2008;359(1):50-60.
Jha AK et al. Use of Electronic health records in U.S. hospitals. N Engl J Med 2009; 360(16):1628-1638.

Non-patient care, billing etc., seems to be in much better shape

Current Electronic (ICU) Systems

- Software Tools
  - Alerts to avoid adverse events
  - Clinical documentation
  - On-line decision support
  - Outcomes tracking
- Emphasis
  - Improving quality of care
  - Remote medicine

Problems of Current Electronic (ICU) Systems

- Alert fatigue
- Doctors don’t control the systems
  - Little customization
- Multiple systems without integration
- Systems don’t bring meaningfull help to the doctors
  - Doctors are only alerted for basic incidents such as values out of range

Listen to Your Doctors Advise

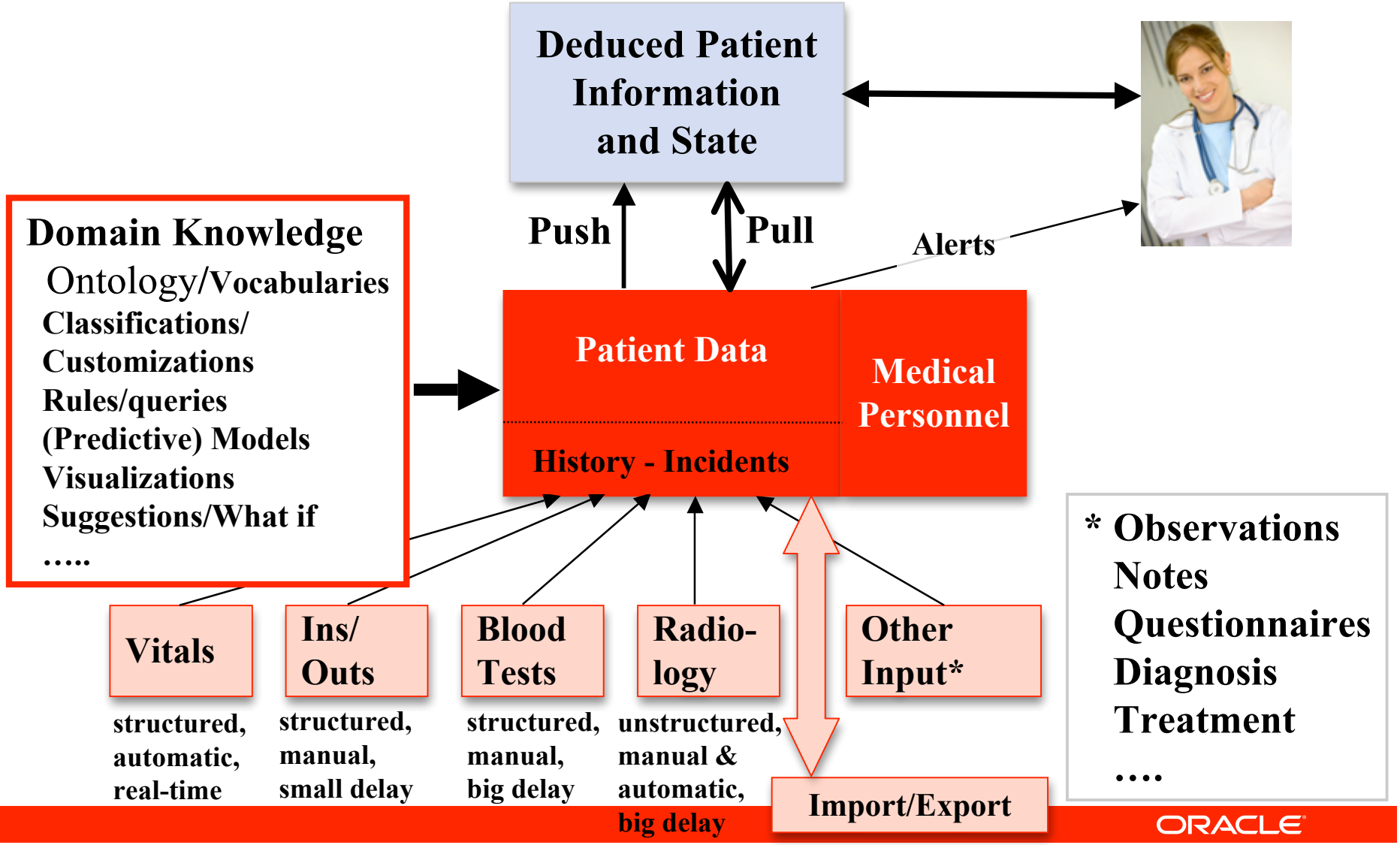
"I'm sitting on a mountain of data hidden behind procedural code"

Dr. Kimball, UUHSC, July 2008

What are Doctors Looking for?

- Support for extracting **evidence** from data
  - Have unobstructed access to all patient data
  - Have unobstructed support for inquiries and analysis
  - Become aware of adverse conditions in a timely fashion
    - As soon as they happened
    - If the likelihood goes beyond a certain threshold
    - Alert based on level of urgency
- Support for **Standard of Care**
  - Organizes alerts
  - Associated meaningful information with any alert

Patient Care – an Overview



We – UUHSC, Coimbra, Oracle - Developed a Prototype

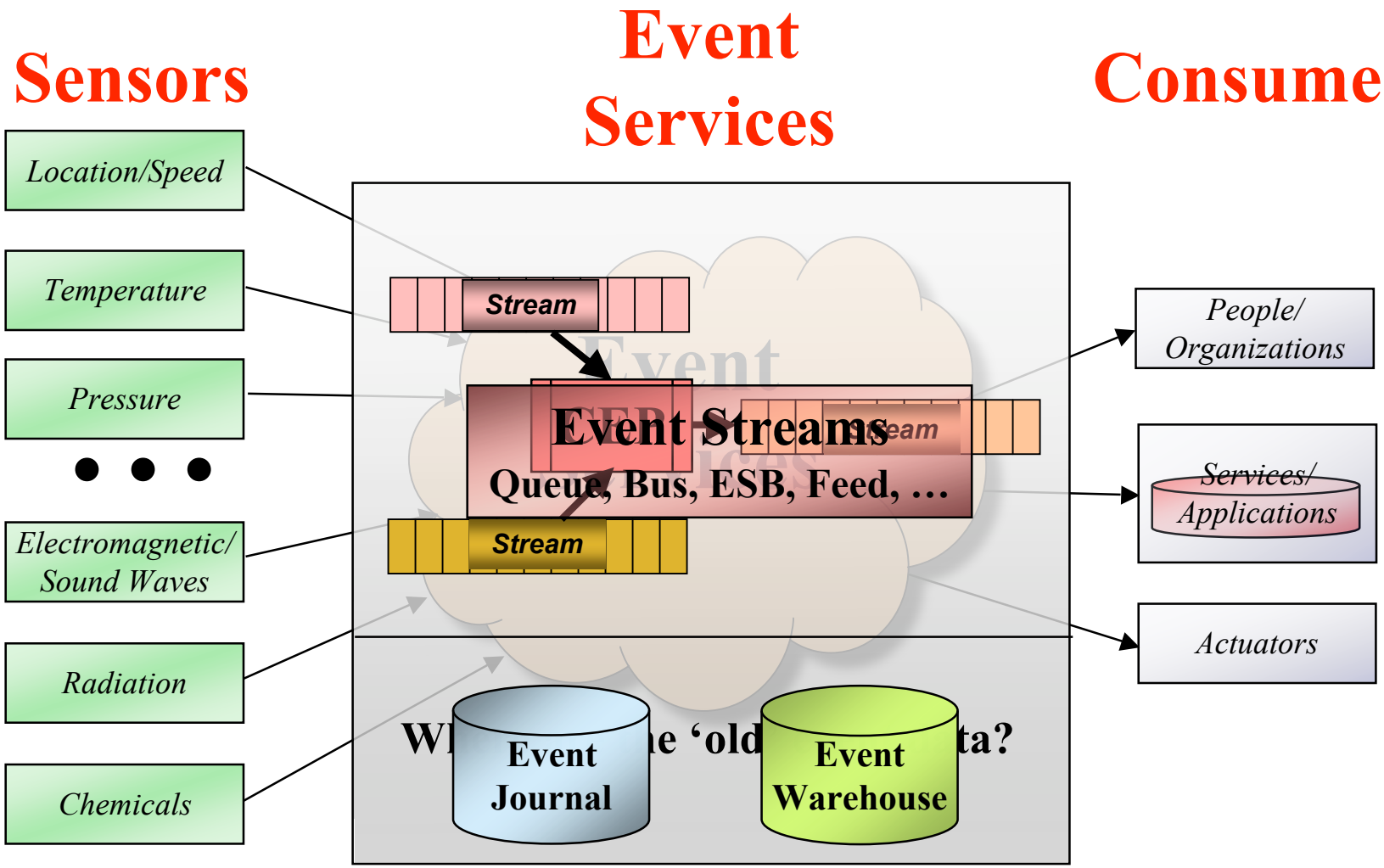
- The prototype provided significant insights in
  - Event processing technology
    - The need for/value of **state based event** processing was recognized
  - A (very) complex, highly customizable, extensible application was developed without using domain specific procedural code
  - Analysis of medical data
    - The idea of 'evidence based medicine' has been pushed significantly ahead
    - *"This (prototype) has the potential to revolutionize medical IT and significantly improve clinical care of a broad range of patients."*

Agenda

- Patient care
  - Exiting IT support for patient care
  - IT requirement for patient care
  - Extracting evidence from patient data
- The technology
  - Continuous queries
  - Issues with CQ technology
  - Registered queries – state based event processing
  - Information and knowledge discovery
- Conclusion

Stream Based Event Processing -

Measurements are captured as Events



Before we go to the Next Slide

Oracle has arguably the most sophisticated/complete technology supporting streams processing

The usual staff as part of a rich infrastructure

It's very fast and very scalable

Pattern detection

- regular expressions, alphabet specified with

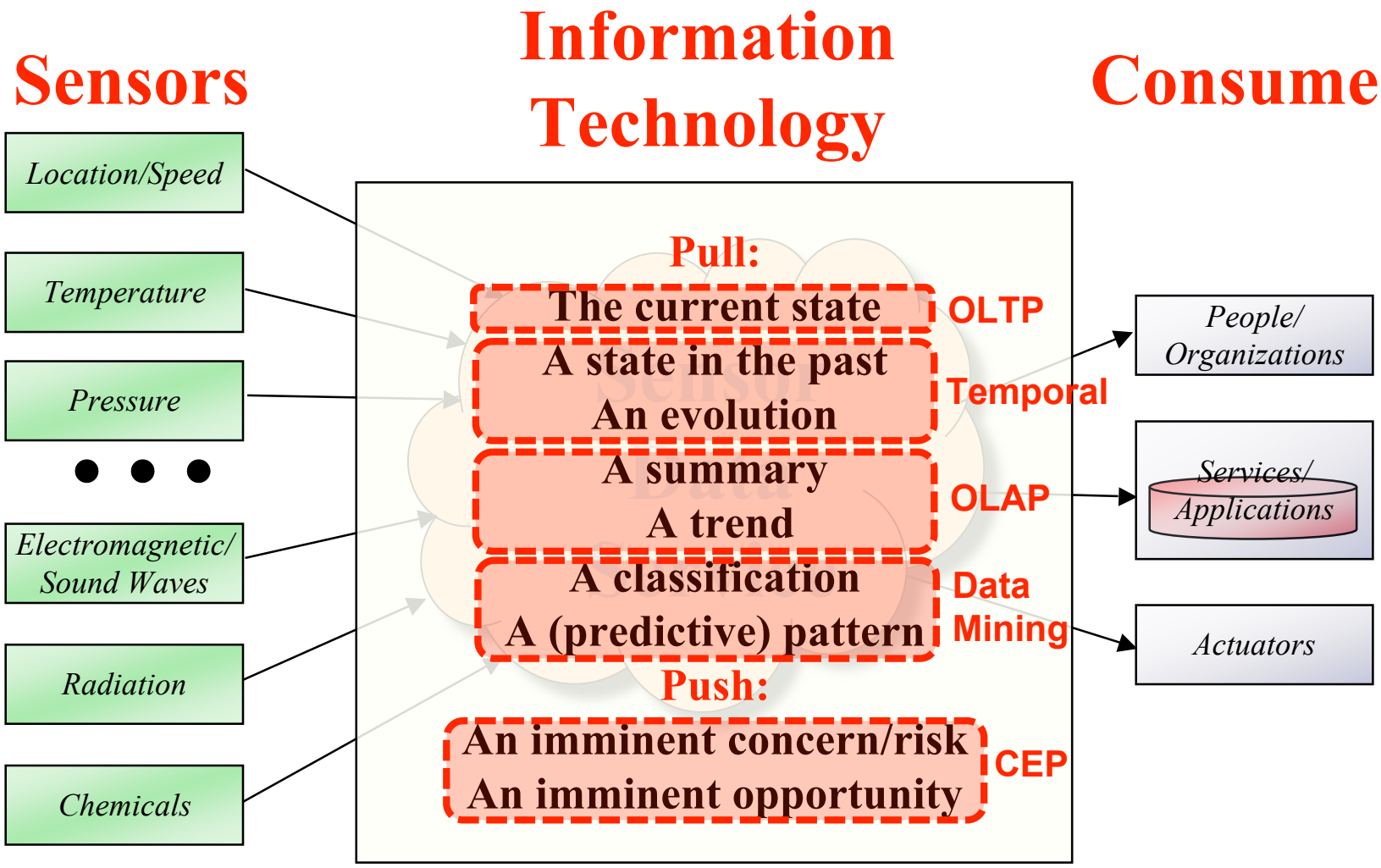
- DEFINED AS –

We can predict what a query returns

Issues with Streams Technology

- Context
  - Sensor data are often only meaningful in the context of other (historical) data
    - Heartbeat of 150
      - Normal for baby
      - Critical for old person
    - Possible solutions
      - Add simple context support to streams technology: typically referred to as ‘cache’
      - Use databases for (complex) context
        - Leads to frequents (pull) access
        - Does not scale in many environments
- Model
  - The data model for events and (historical) data is inconsistent
    - Leads to confusion

State Based Event Processing –
Measurements are Captured as Facts



Oracle Database Server Support

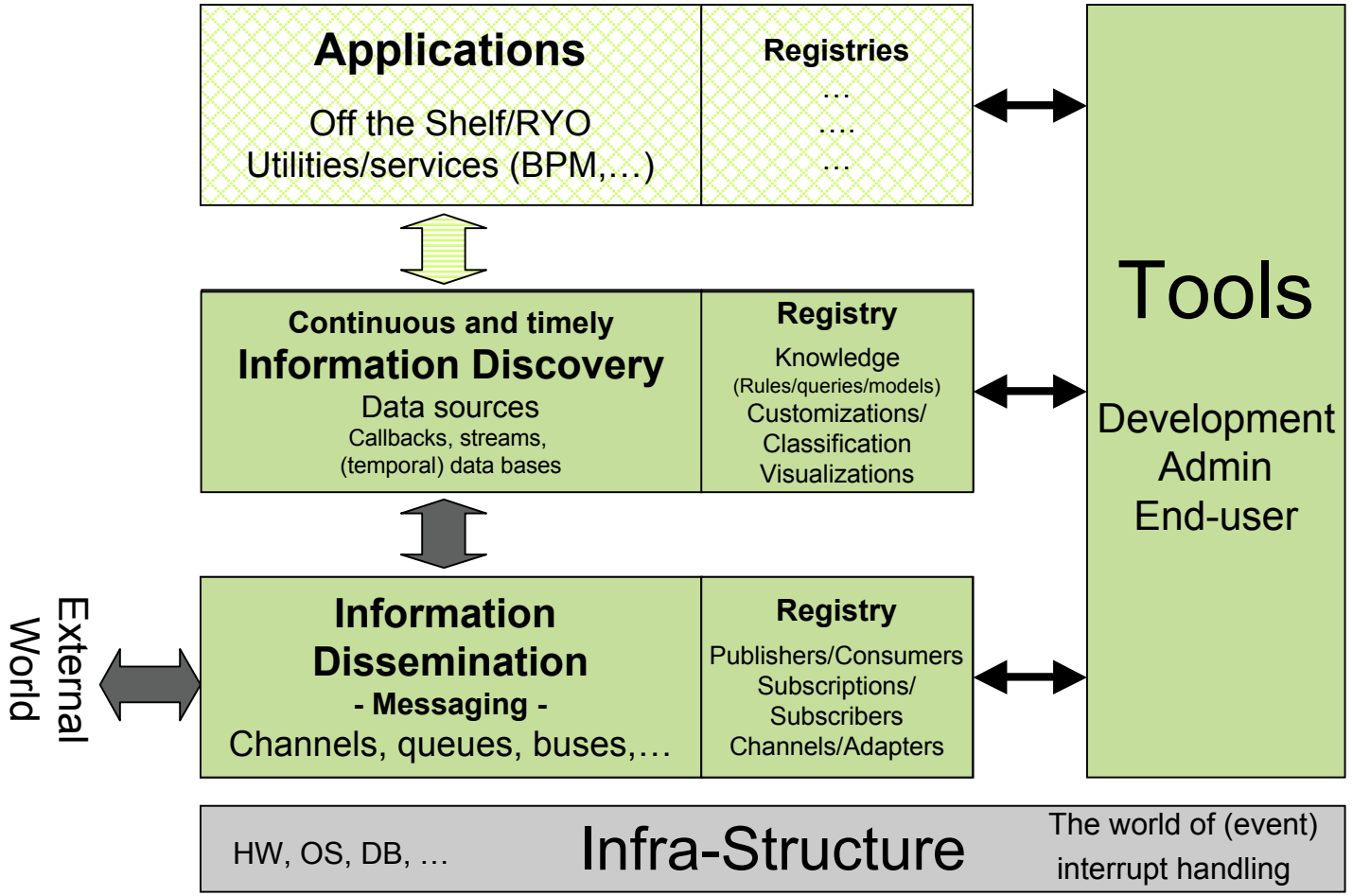
- **Total Recall** (TR) – Transparent versioning of records. Access to historical data
- **Continuous Query Notification** (CQN) – Notification system for committed data
  - Standard queries plus destination – **registered queries**
    - Applies to SQL, XML, SPARQL/OWL, extensibility
  - Notification on change of result set
- **Rules Manager** (RM) – Rules engine. Allows pattern matching, complex alerting, etc.
- **Oracle Data Mining** (ODM) – Data mining engine embedded in the database. Allows scoring directly from database data

Information and Knowledge Discovery

- Timely information discovery
  - Continuous analysis of data for timely discovery of domain relevant information using rules/queries/models
  - Capturing relevant information in 'incident objects'
    - Events are the artifact creating and managing incident objects
- Knowledge discovery
  - Discovery (and verification) of rules and models for improved information discovery

- Continuous/registered queries provide timely awareness of 'evidence'
  - Expert system level: Simple/complex rules/queries
  - 'Post' expert system level: Permanent scoring of non-hypothesis driven models

Information Discovery as Critical Part of the 'Management of Data' or a History of Event Processing



Agenda

- Patient care
  - Exiting IT support for patient care
  - IT requirement for patient care
  - Extracting evidence from patient data
- The technology
  - Continuous queries
  - Issues with CQ technology
  - Registered queries – state based event processing
  - Information and knowledge discovery
- Conclusion

Conclusion

- We – the database community - should think about a new approach to (database) applications
  - Data capturing – we are good at this
  - Continuous and timely information extraction – this is what we are focusing on
    - Meta data driven (ontology, rules, models, …)
    - Based on continuous/registered queries
    - Minimal or no procedural code
  - Standard reaction – we need to come up with new ideas
- Types of event processing
  - Streams based - CQL
  - State based - Registered queries