

Sedvice: Triple-Space Computing Exploration Platform

Ian Oliver

Tripcom Workshop

Galway

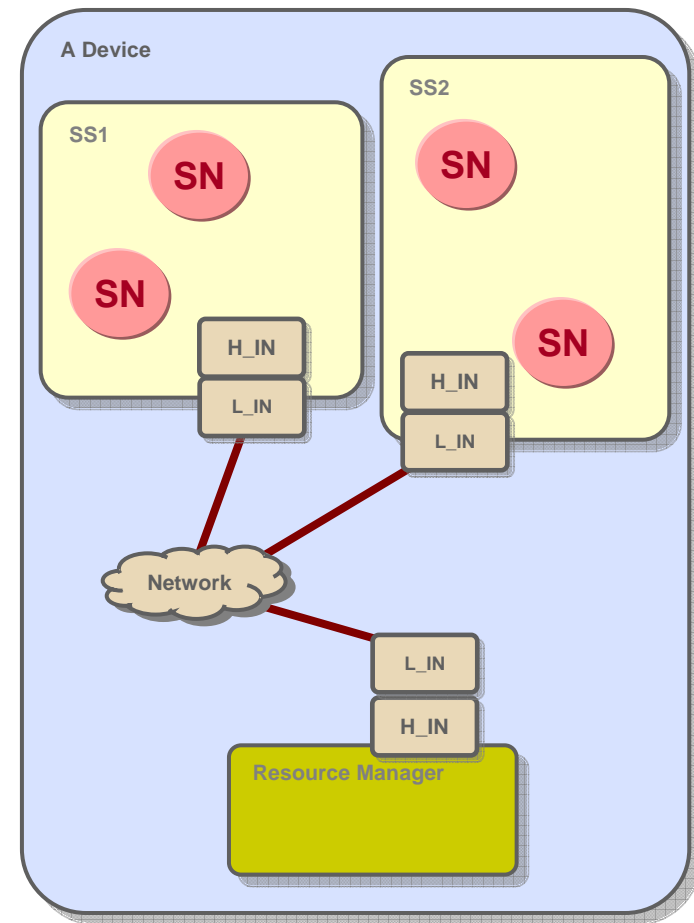
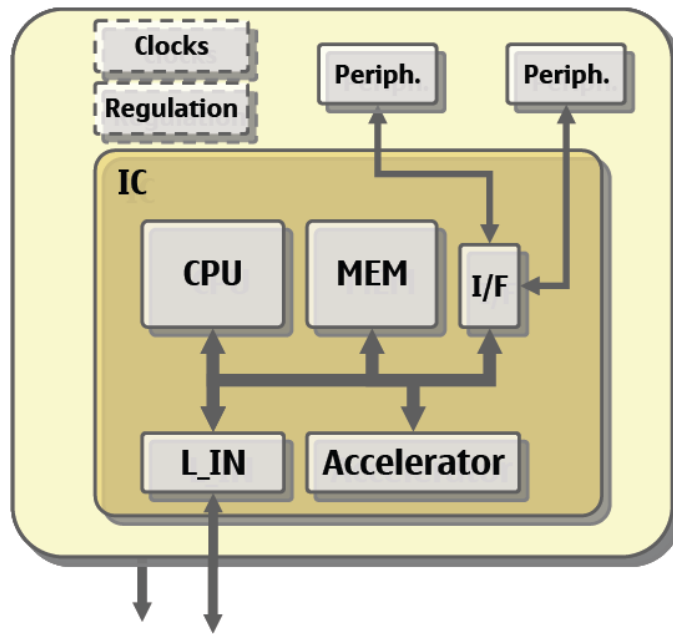
29 April 2008

Aims

- Investigate new/novel mobile-device architectures
 - On and off device
 - hardware/software
- Specific focii
 - universal connectivity
 - interoperability (via the Semantic Web)
 - modularity
 - power consumption
- Buzzwords
 - Web 3.0
 - **Dynamic Adaptive Semantic Web**™

Pre-History (2004-2006)

- Network on Terminal Architecture (NoTA)



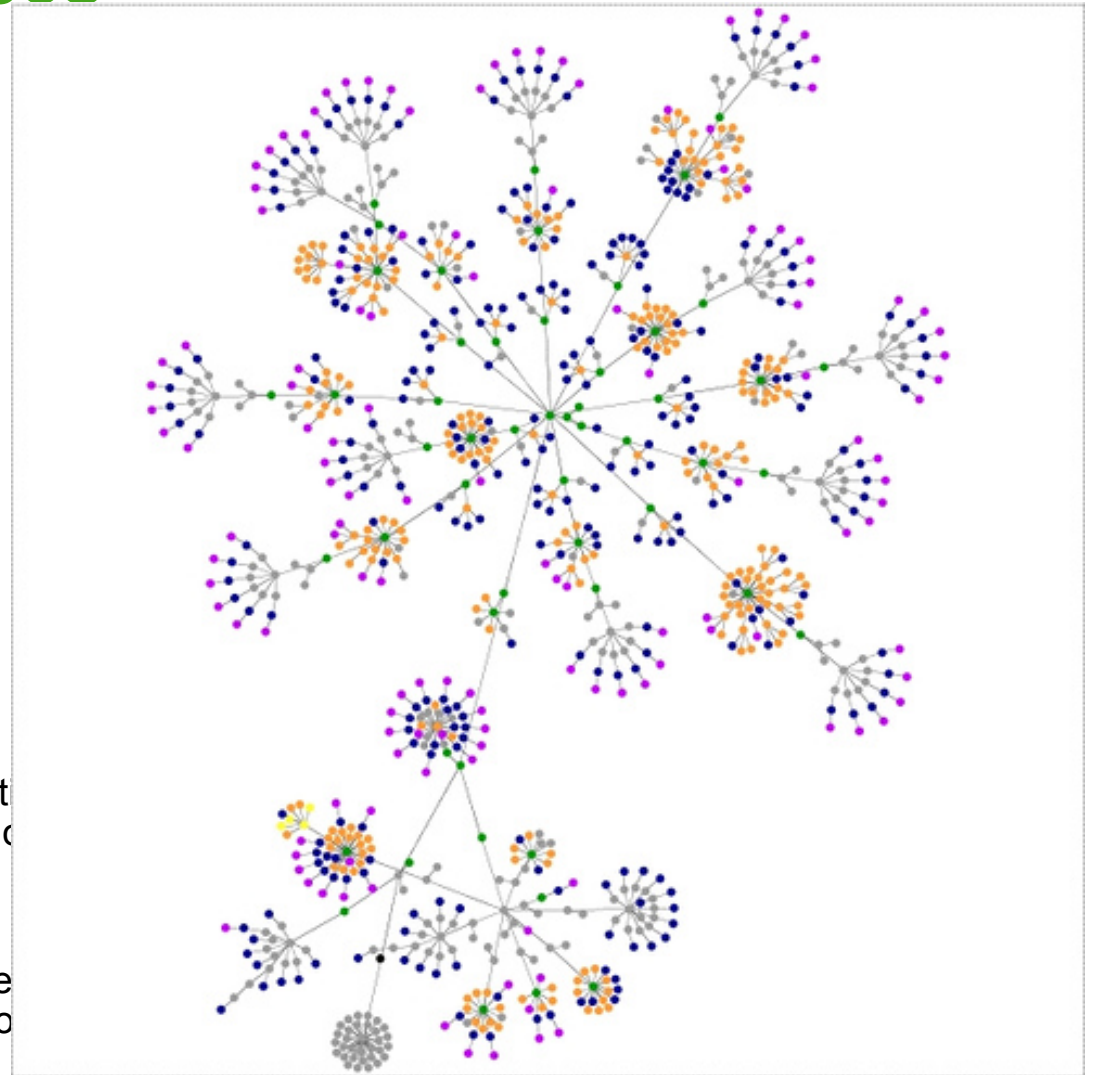
History (2006-2007)

- Sedvice/M3
 - interoperability (via semantic web constructs)
 - architectures
- Triple-Space based distributed computing platform
- Built upon Wilbur
- move to investigate distributed computing



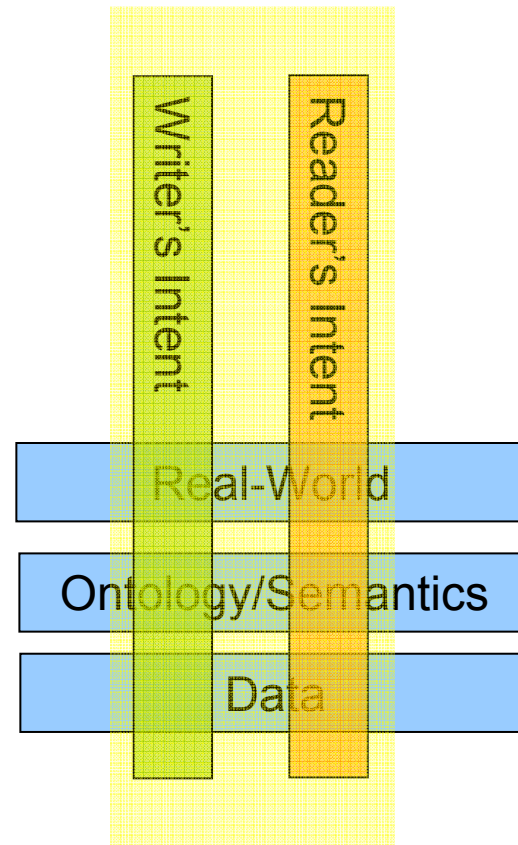
The Present - SmartSpaces

- Nokia Triple-Space Based Computation Platform (Sedvice)
- **A Localised Information Sharing Mechanism**
 - Abstraction from the device
 - Abstraction from control flow
 - Abstraction from services
 - Abstraction from application
- An information centric environment for the declarative construction of applications from the manipulation of information
- An environment which allows real-time usage of semi-structured information with an emphasis on intention semantics



Information

- Data is expressed as a directed graph structure, ostensibly represented in RDF
- Ontology provides structure and a default semantics to this
- The semantics of the ontology is grounded in real-world concepts
- However, any piece of information has semantics given by the creator of that information which is the intent of that information
- Which is further complicated by the fact that some writers might ascribe a different intent to that information
- Which if you're lucky intersect such that the intent of both parties is understood



Sedvice Architecture

- Spaces

- represented by one or more SIBs (Semantic Information Broker)
- Contains
 - membership & policy management
 - distribution protocols (internal distribution)
 - information processing and rewriting
 - information storage (Wilbur)
 - may be connected to external storage directly

- Nodes

- actors/agents
- autonomous, concurrent

- Communication

- information based
 - structured as RDF
 - not considered "data"
- no control flow
 - may be made "outside" of the system via NoTA, UPnP, Webservices etc
- semi-structured information
 - no strict ontology conformance
 - inconsistent information allowed!
- semantics of information, belief and truth maintenance responsibility of the reader (agent/actor)
- free logics
 - depends upon the Space, but 2-valued, 3-valued, 4+
 - modal, subjective, classical...
 - notions of unknown (\perp) are probably desirable

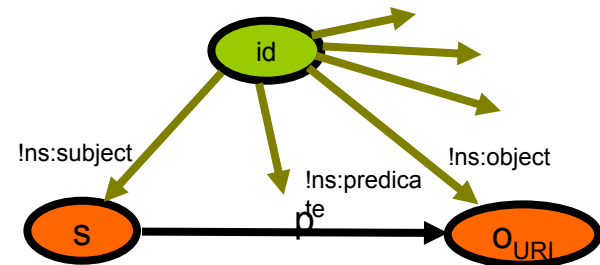
Functionality

- Basic functionality
 - insert }
• retract } transactional
 - *non-monotonic mode of operation only*
 - query (synchronous)
 - set/bag, full results/announce
 - subscription (asynchronous)
 - set/bag, full results/delta/announce
 - WQL
 - deductive closure calculation
 - Direct RDF triple queries
 - no reasoning
- Formal Specification over Architecture using Alloy
- Policy/Security
 - join/leave
 - invite/remove (real soon now)
 - information based policies (2009)
- Advanced
 - triple rewriting and optimisation
- Implementation/Productisation
 - Python 2.5.1...Symbian from 2H2008
 - C++/OpenC core due 2H2008
 - Client libraries
 - Python, C, OpenC 1H2008 (support for Flash)
 - C++, Java, 2H2008

Triples++

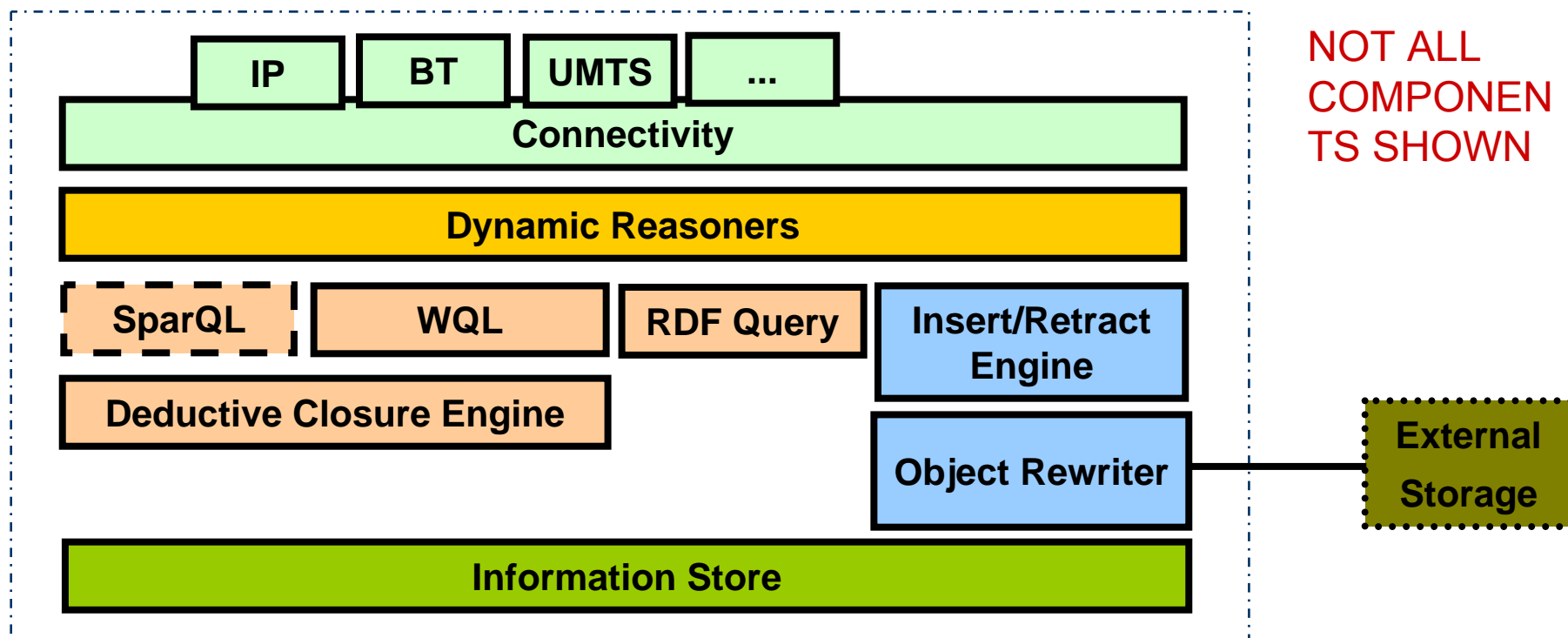
- Triples are input as (s,p,o) but are stored internally as (s,p,o,*ptr)
- The *ptr is a URI which points to a structure which may contain whatever required information is necessary
 - typically:
 - meta-data
 - source
 - etc...
- *ptr may be NULL

- Triple reification
 - possible but generally switched off
 - generates too many additional triples

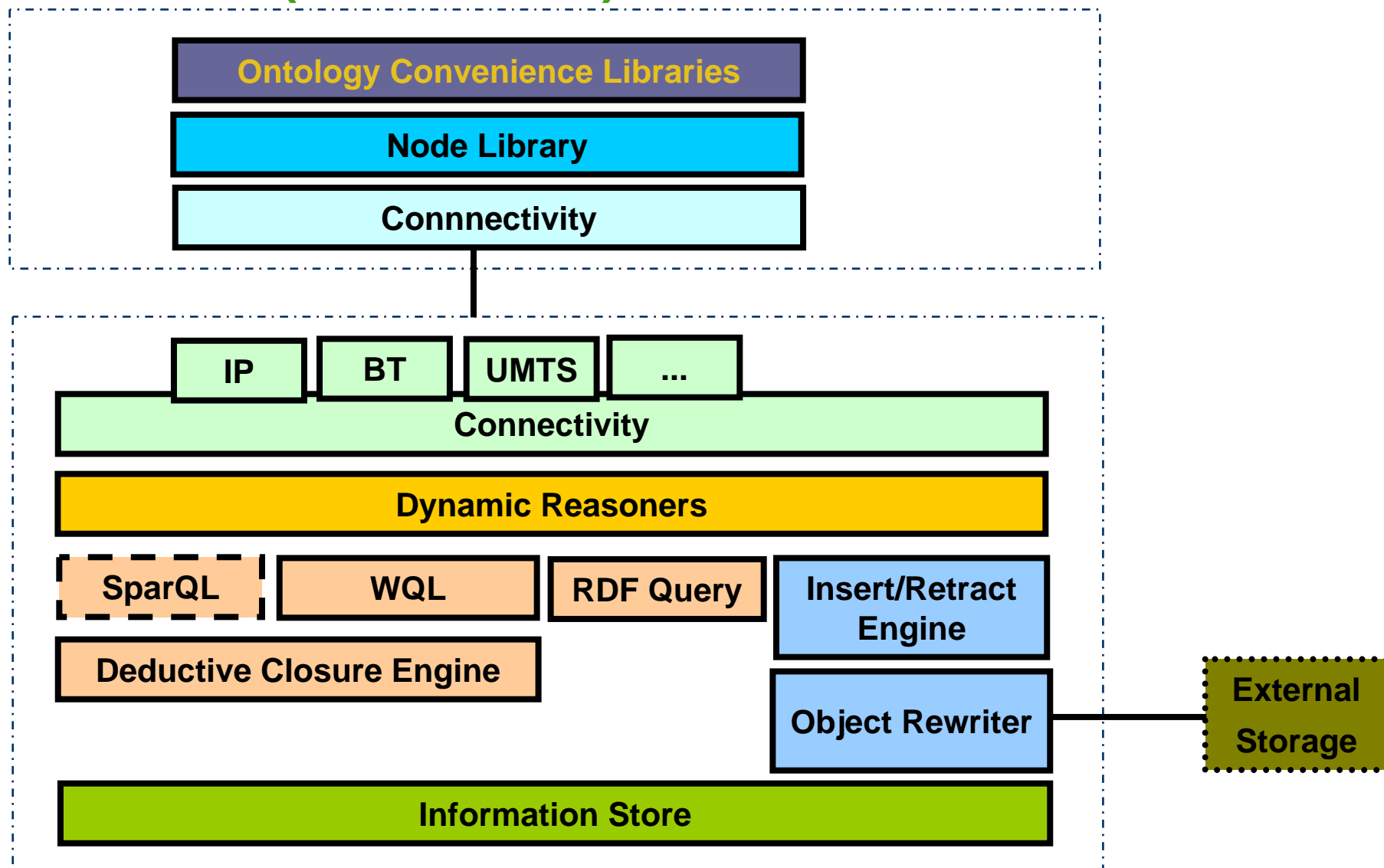


- not elegant enough... *ptr does it better
 - 400-500% overhead
 - not everything generates triple-Ids
 - semantics is awful
 - literals are !rdf:properties?

Pictorial View (SIB)

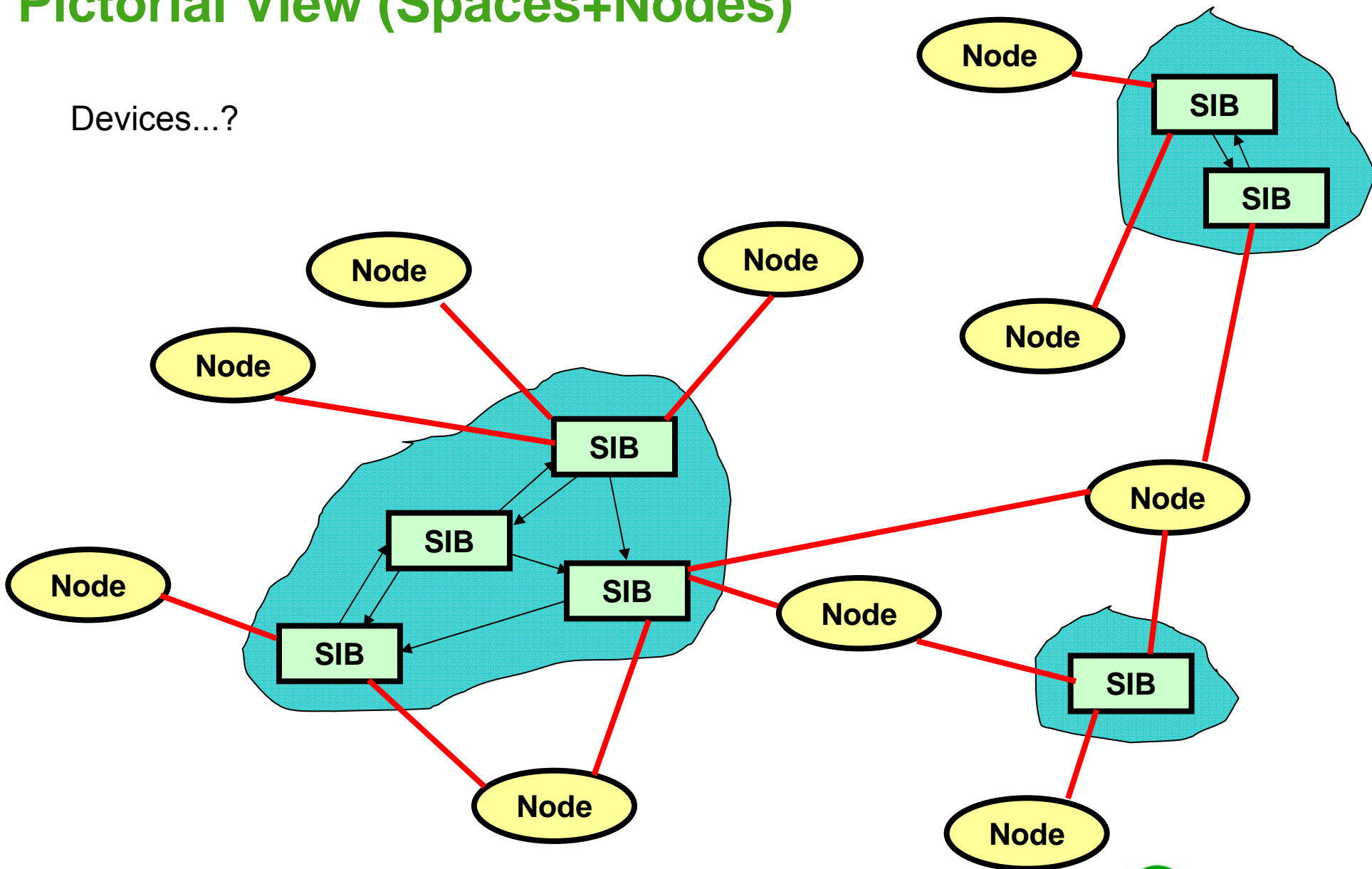


Pictorial View (SIB+Nodes)



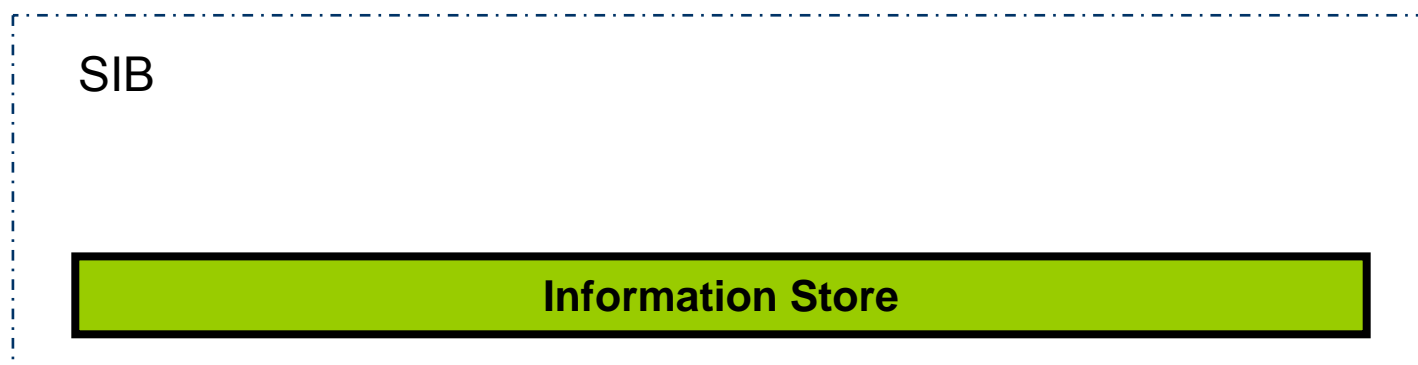
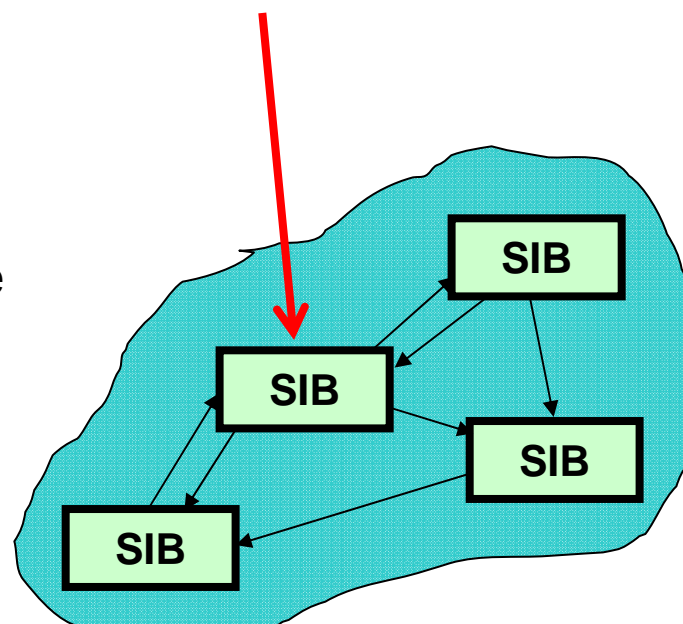
Pictorial View (Spaces+Nodes)

Devices...?



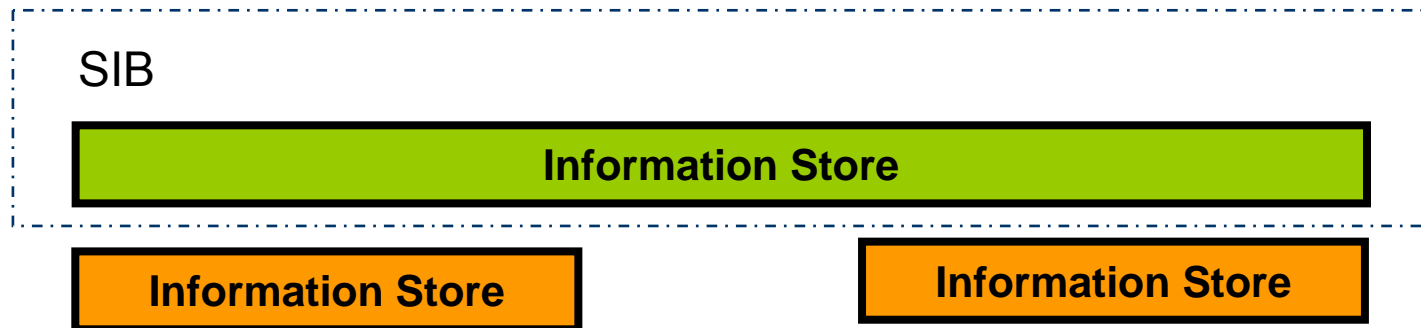
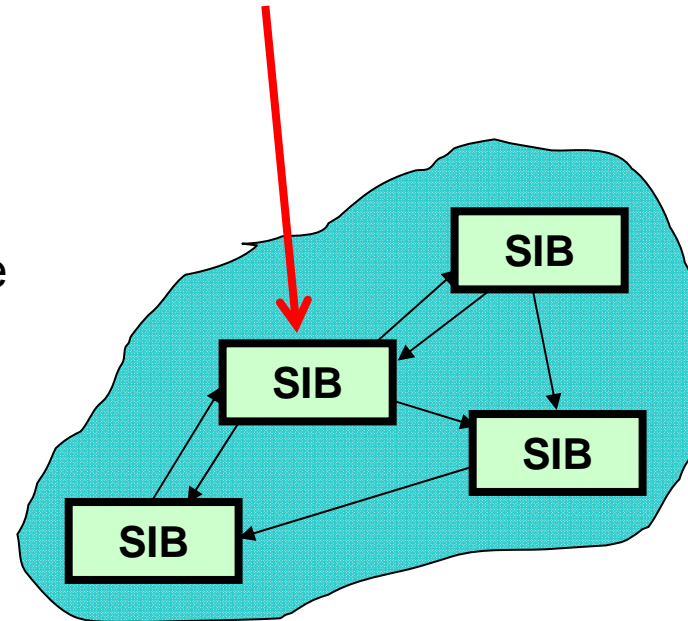
Space Distribution

- Spaces are self-contained
- Information may be distributed amongst the SIBs within a space
 - distributed insert/retract semantics
 - distributed query semantics



Space Distribution

- Spaces are self-contained
- Information may be distributed amongst the SIBs within a space
 - distributed insert/retract semantics
 - distributed query semantics
 - deductive closure calculation:
 - $D(S_1(Q) \cup S_2(Q)) \neq D(S_1(Q)) \cup D(S_2(Q))$

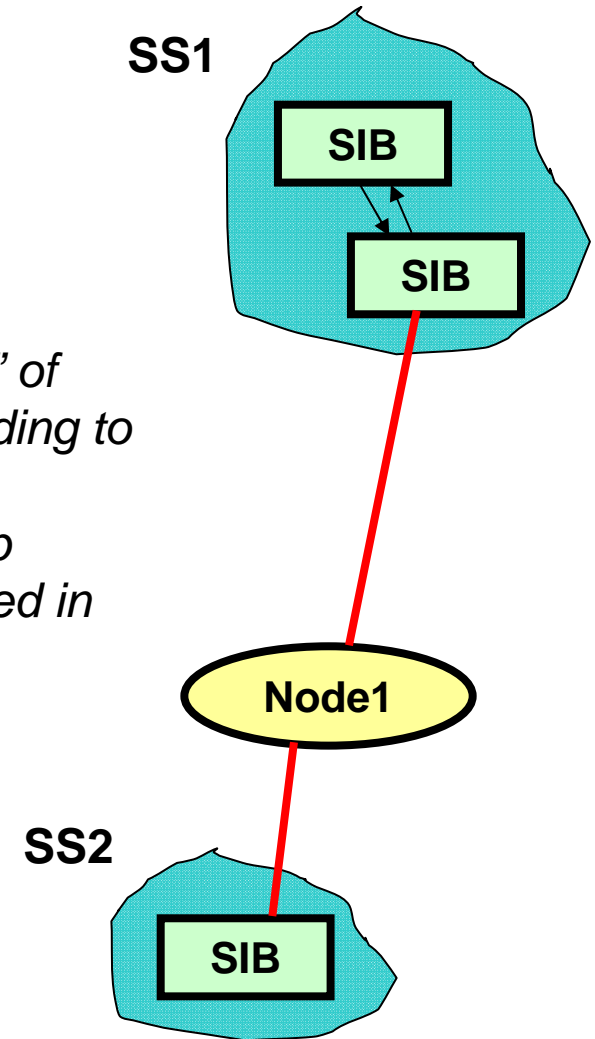


- Distributing the information store may be made...not currently considered - databases do this so much better!

Space Relationships

- Subspaces, space relationships were removed as being "first class" elements
- Can be "lifted" into Nodes
- Good features:
 - any relationship can be implemented
- Bad features:
 - cyclic relationships
 - harder to ensure relationship semantics

SS1 is a "subspace" of SS2 according to some relationship implemented in Node1



Security, Policy & Trust

- Not implemented !!!
 - not yet!, 1H/2H 2008ish...
 - privacy is over-rated anyway 😊
 - major concern and requires a lot more research
 - unclear at this time how these are going to be implemented
- Join/Leave mechanisms
 - Node -> Space request
- Invite/Remove mechanisms
 - Space -> Node request
- Security
 - definition of ?
- Policy
 - triple/uri based
 - larger structures?
 - DL based + reasoning mechanism
- Trust
 - not considered in detail
 - subjective logics

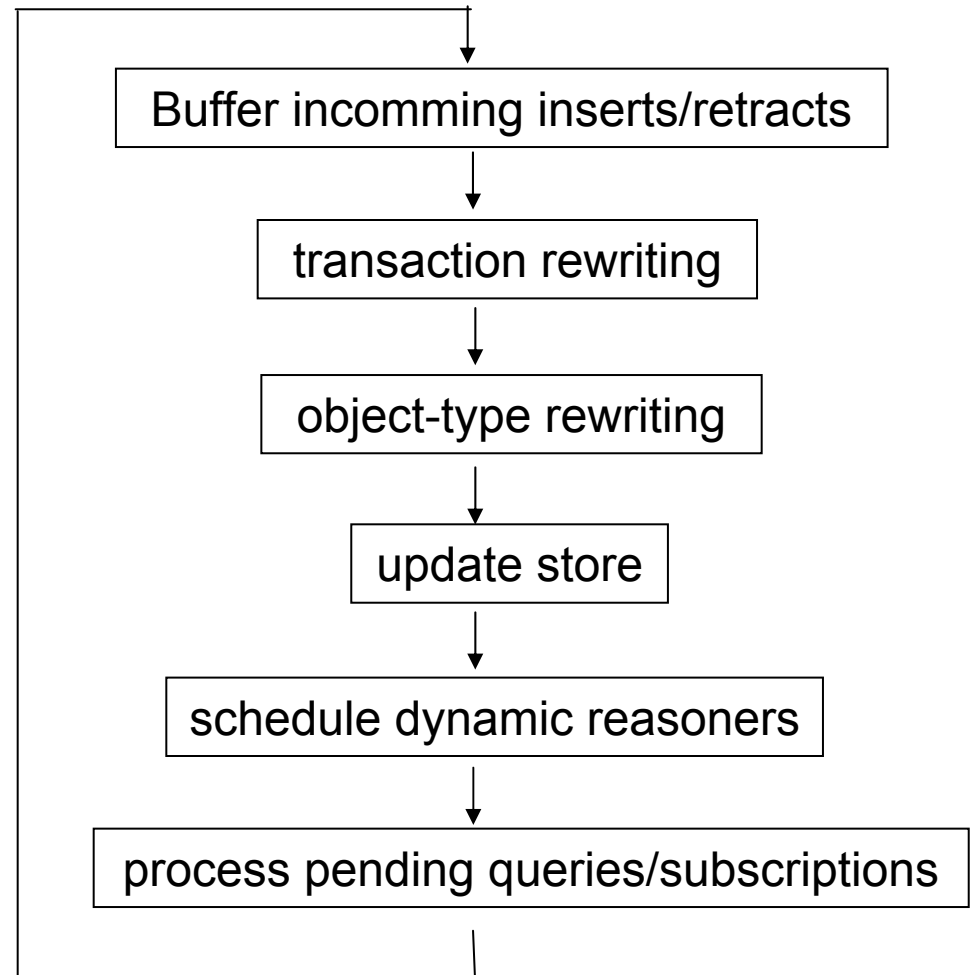
Static Reasoning and Triple Rewriting

- WQL queries utilise the calculation of a deductive closer dynamically
 - no static creation of triples that can be inferred
 - `rdf:type` under `rdf:subClass`
 - `rdfs:Property`
 - `owl:sameAs`
- Triple rewriting
 - object type inference
 - object type rewriting
 - BLOB, large object optimisation
 - off-Space storage
 - FLICKR, YouTube...
- Transaction rewriting
 - $\{\text{insert}(a), \text{retract}(a)\} \diamond \sim = \emptyset$
 - plug-in logic system
 - insert and retract are transactional
 - transactional = multiple `i/r`'s atomically
 - all operations are atomic invidiaully
 - query and subscribe aren't – you get whatever

Dynamic Reasoning

- Internal Nodes to a Space
- Typical functionality
 - Ontology Adherence
 - Ontology Repair
- Differences from Nodes
 - guaranteed execution
 - easier to the mathematics/semantics
 - scheduling environment
 - specific ordering of reasoners

- Scheduler



Applications

- Semantic Web has no killer application
- Semantic Web is the killer application
- Demo applications
 - Chat
 - investigations into coordination, distribution and general application design
 - MediaPlayer
 - canonical Nokia application
 - UPnP Integration
 - UPnP CDS + device + family integration
 - "family" UPnP
 - Health and Wellbeing
 - Nokia SportsTracker
 - Location
 - GPS integration
 - Personal Media
 - photographs + identity + location + external storage
 - context collaction
 - NLP interfaces
- What is an application?
 - not monolithic
 - distributed
 - distribute over functionality?
 - control flow/coordination?
- Ontology, ontology, ontology, ONTOLOGY!
 - "ontologisise" everything...

Demo Setup

- 1xWindows PC
- 1xLinuxPC
- 1xN770 (Linux)
- 3xN800(Linux)
- 1xN95 (Symbian, Java)
- 1xSolaris Server (not shown)

- AdHoc Network (+ some routing to internet)
 - WLAN/IP Based
- SIB running on N800 (centre picture)
 - python version

- 2 Spaces + Nodes including:
 - Chat (approx 10 Nodes, 3 humans)
 - Weather aggregator
 - Chat-Weather interface

- Everything Python (2.5.1) except hand written, adhoc, MIDP Java implementation



Ontological Considerations

- Everything needs to be defined via ontology
 - though no strict adherence to the ontology is required
 - inconsistency and abuse of meta-levels, typing etc is fine
 - see RDF Model Theory for a discussion of ZF axioms
- Application Development
 - ontology centred
 - automatic convenience library generation
 - distribution of functionality
 - no control-flow
 - algebraic structures/lambda calculus/pi-calculus
- Scope Rules
 - embedding query (WQL) in the RDF store
 - dynamic/on-the fly convenience library generation
- Reflection
 - internal processing of scope rules
 - embedded execution
- Retraction
 - notions of aggregation/composition
 - cf: UML