Semantic computation and the Future Semantic Web

Ian Oliver
Nokia Research Centre
Helsinki, Finland

FRUCT 2008
30 October 2008
Tampere, Finland

5 September 2008

Nokia Research Center

NOKIA

Contents

Part one:

Ancient History, Present and Future

- Web to Semantic Web
- Applications, Pages and Agents

Part two:

Technologies

- Semantic Computation
  - Spaces and Agents
  - Scalability
  - What are Applications?
- Theoretical Underpinnings
  - Graphs, Graph Structures
  - Agents, Spaces
  - Ontologies, Semantics
  - Total Abstraction
- Demonstration

Part One

Nokia Research Center

NOKIA

History

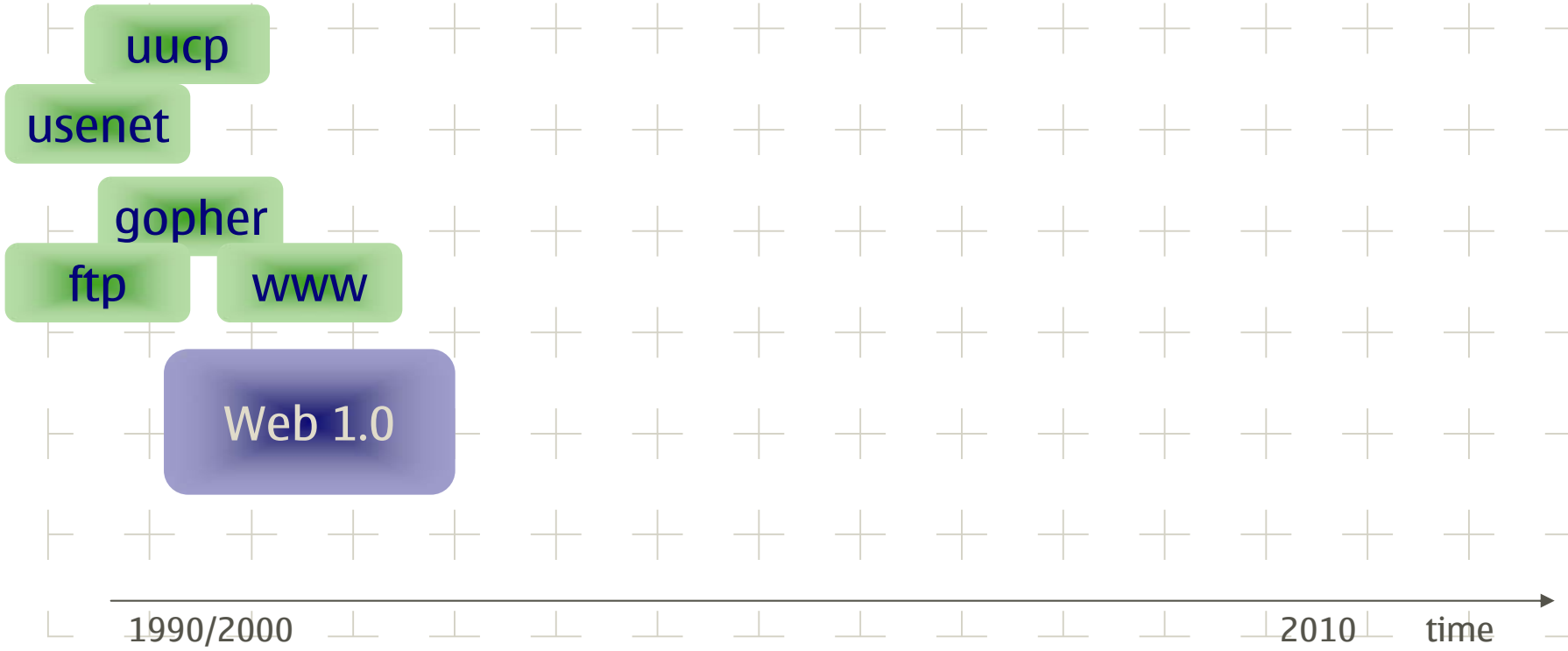
uucp

usenet

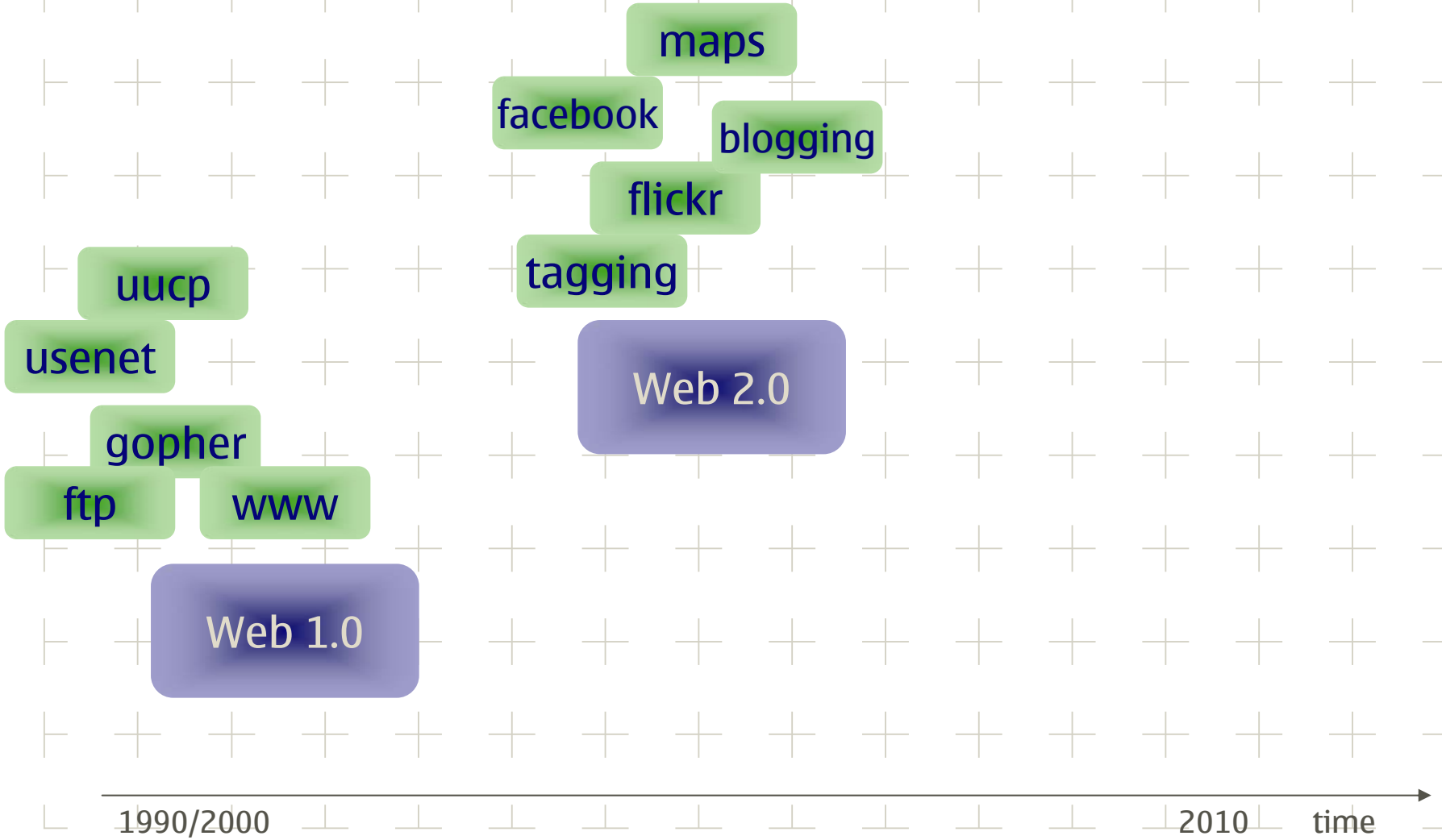
gopher

ftp

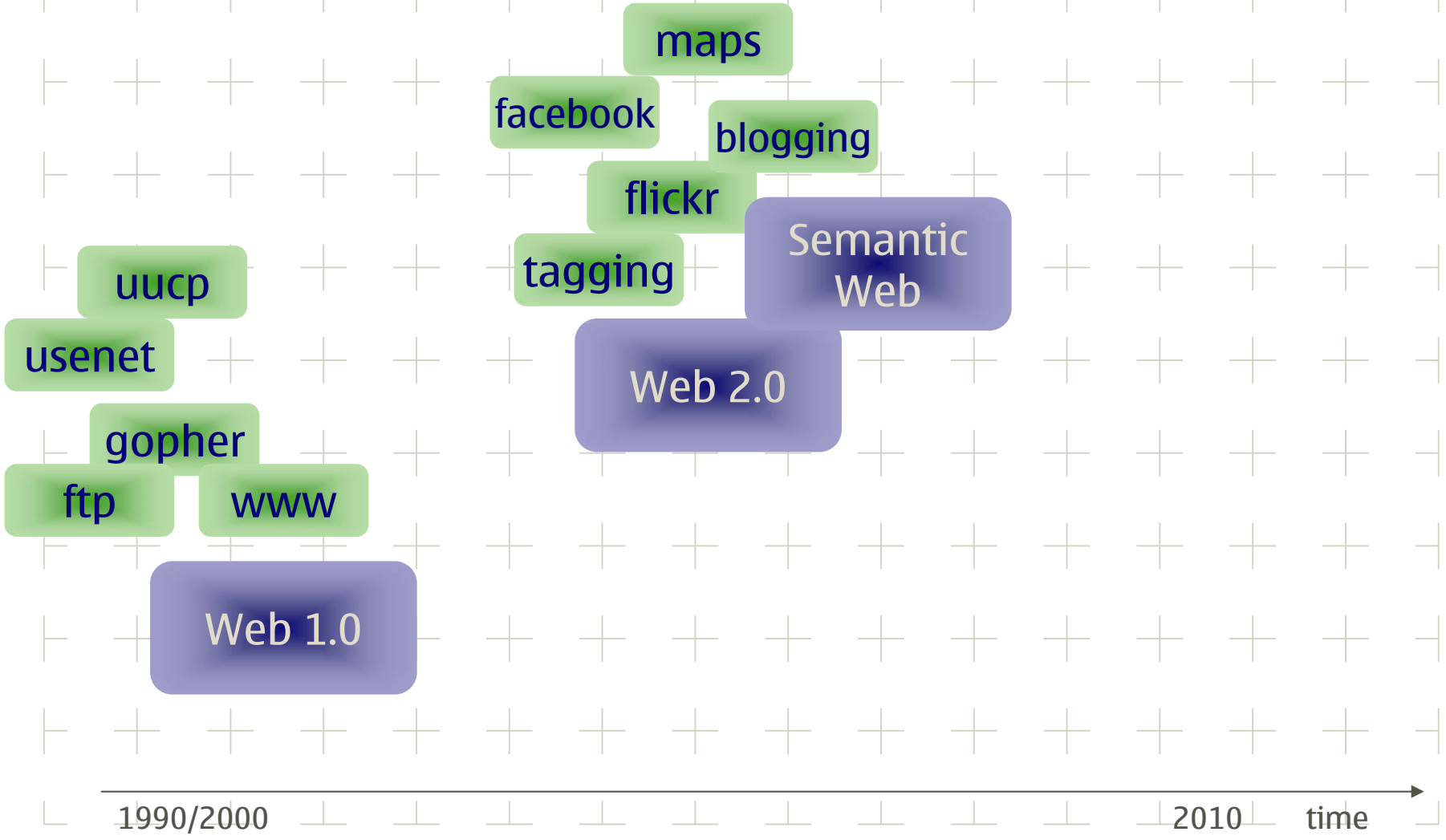
History



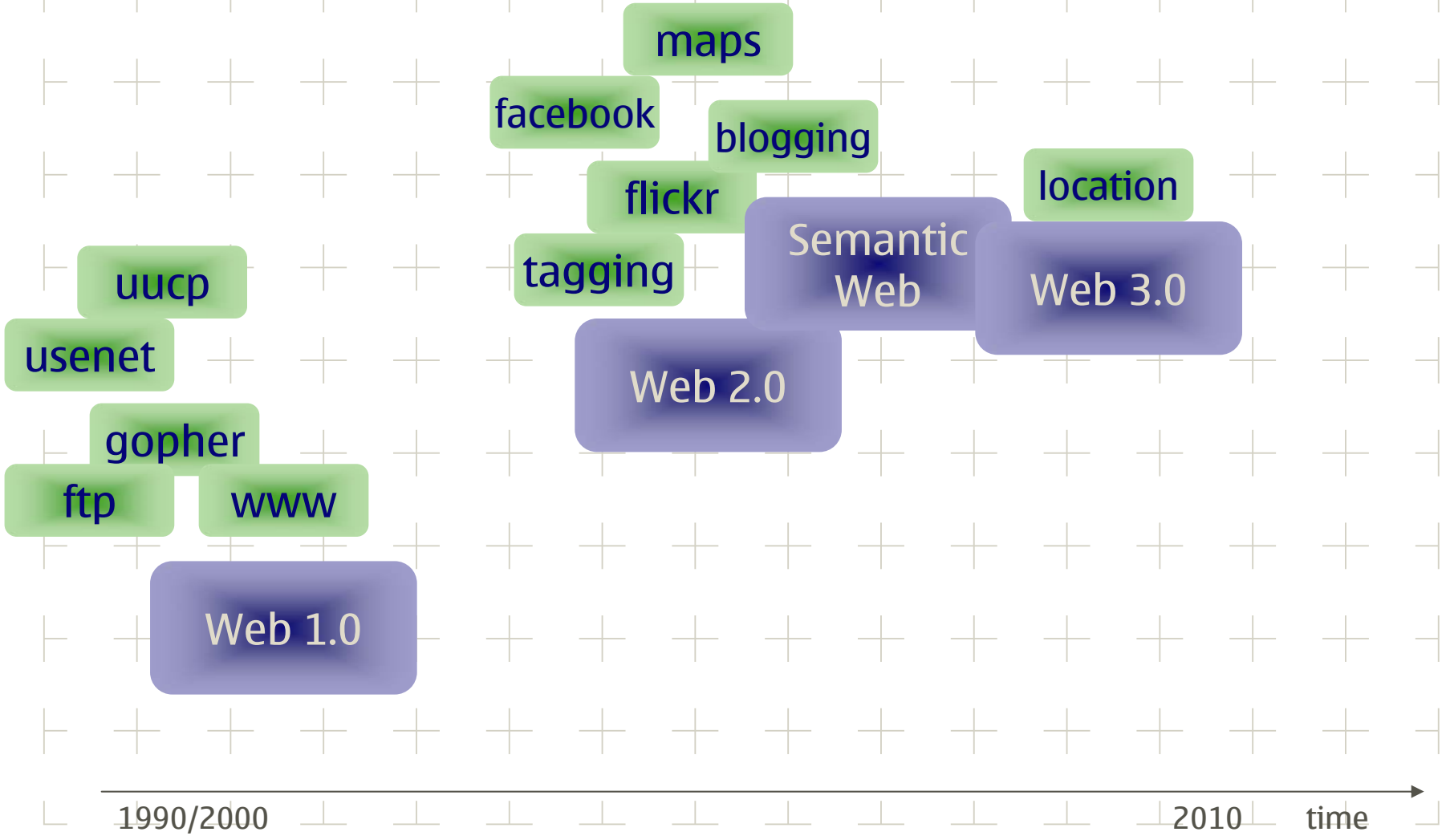
History


maps
facebook
blogging
flickr
tagging
uucp
usenet
Web 2.0
gopher
ftp
www
Web 1.0
1990/2000
2010
time

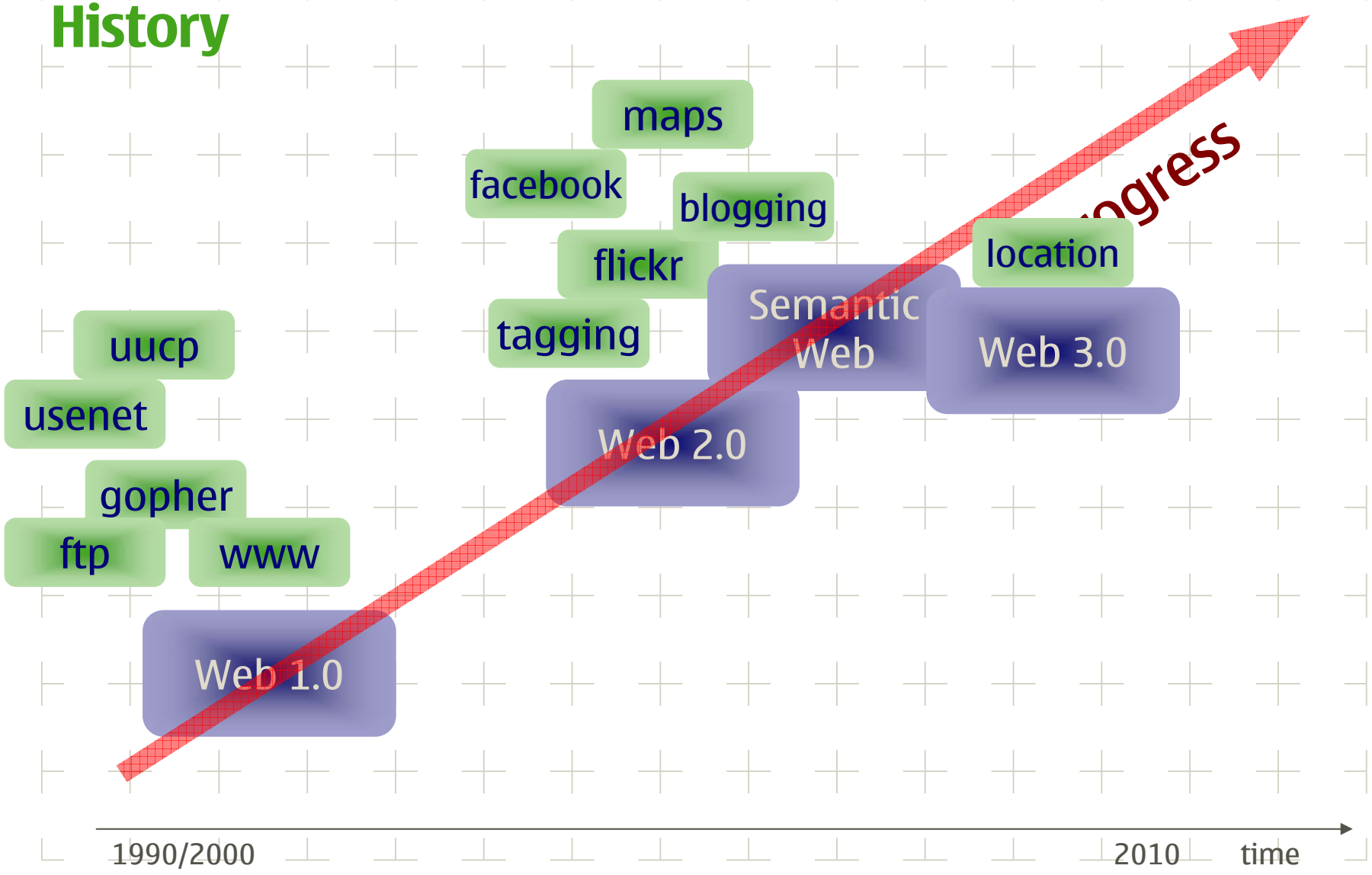
History


maps
facebook
blogging
flickr
tagging
Semantic Web
uucp
usenet
Web 2.0
gopher
ftp
www
Web 1.0
1990/2000
2010
time

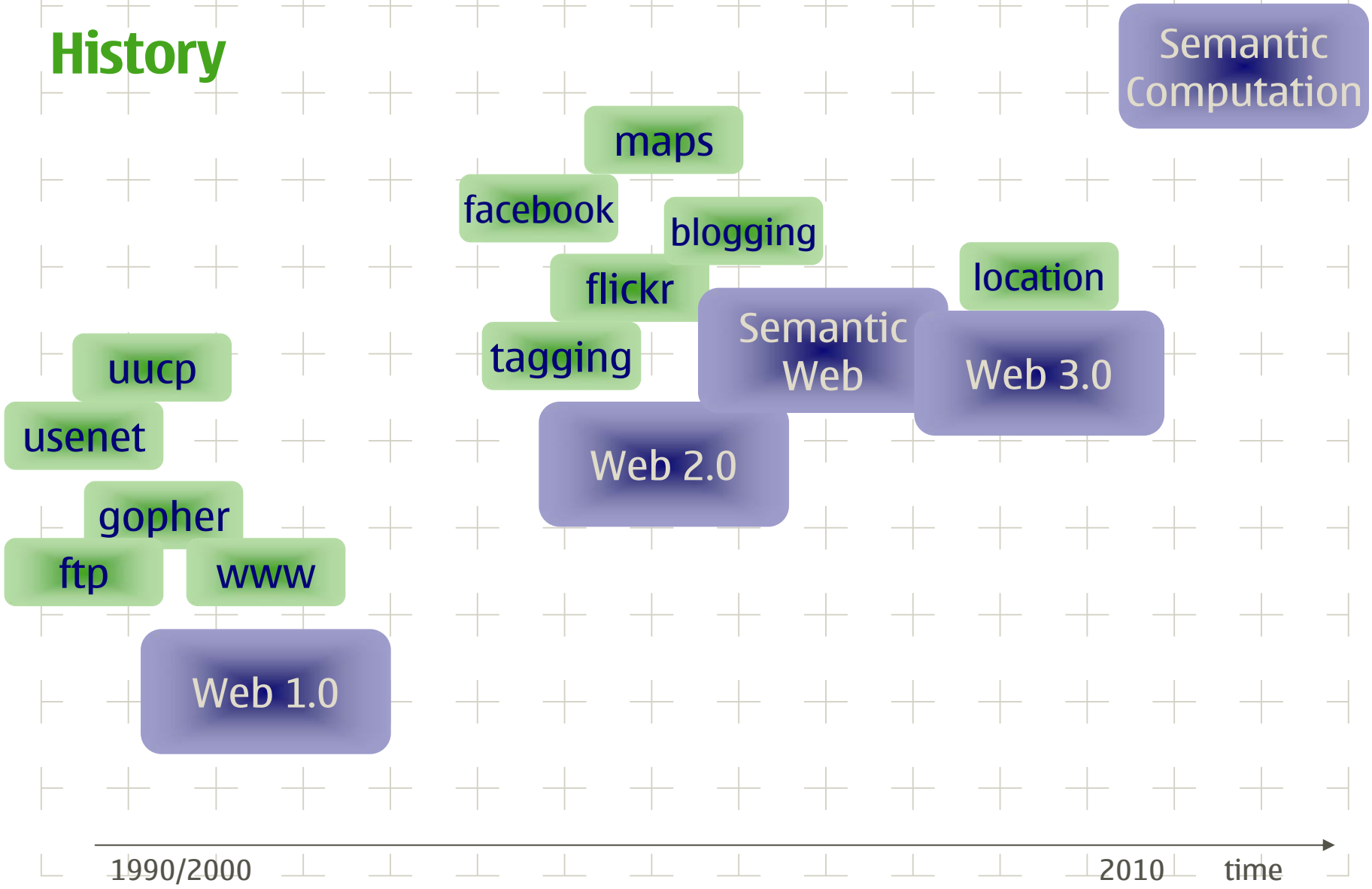
History


maps
facebook
blogging
flickr
location
Semantic Web
tagging
Web 3.0
uucp
usenet
Web 2.0
gopher
ftp
www
Web 1.0
1990/2000
2010
time

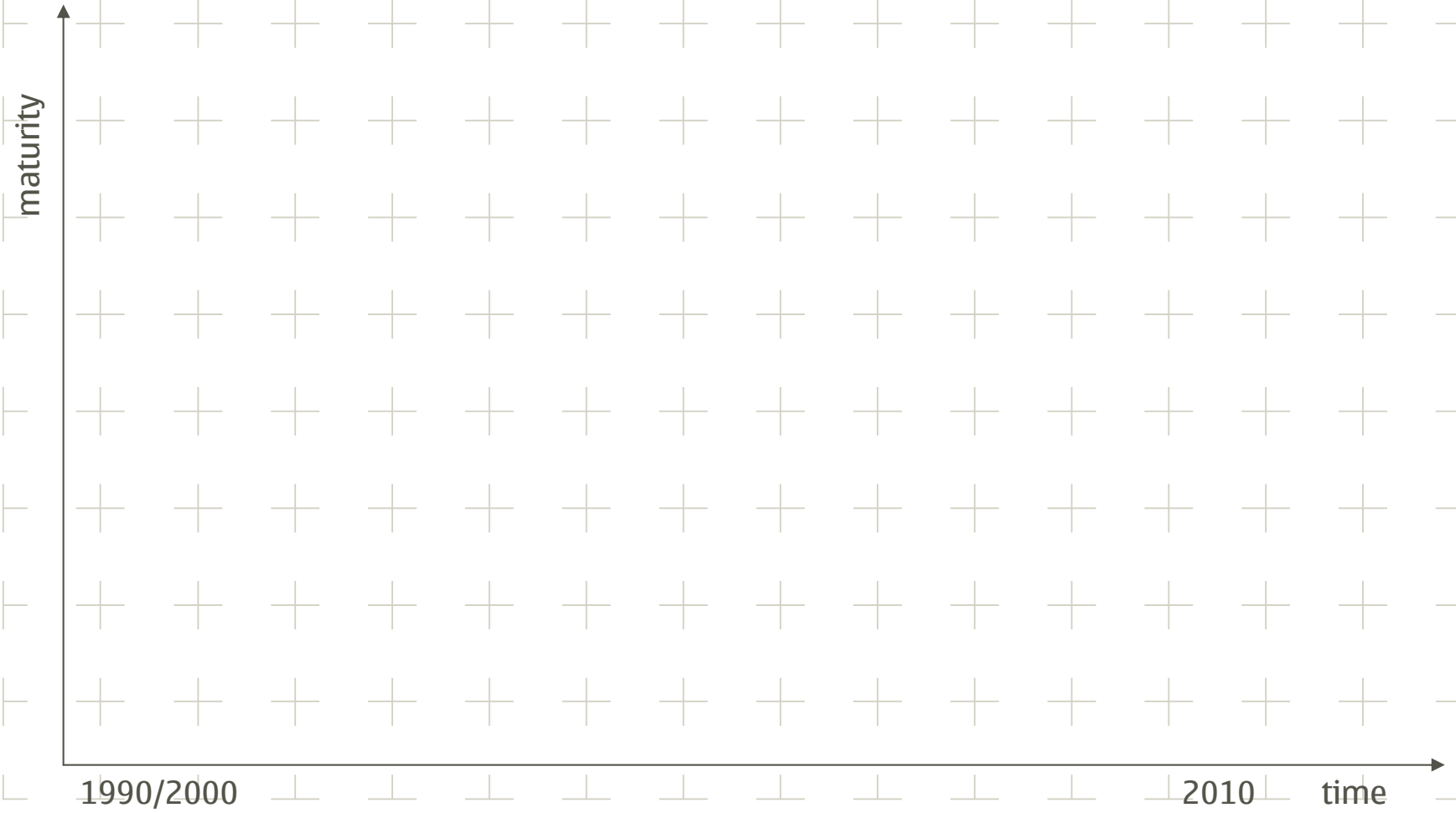
History


maps
facebook
blogging
flickr
location
tagging
Semantic Web
Web 3.0
uucp
usenet
Web 2.0
gopher
ftp
www
Web 1.0
1990/2000
2010
time

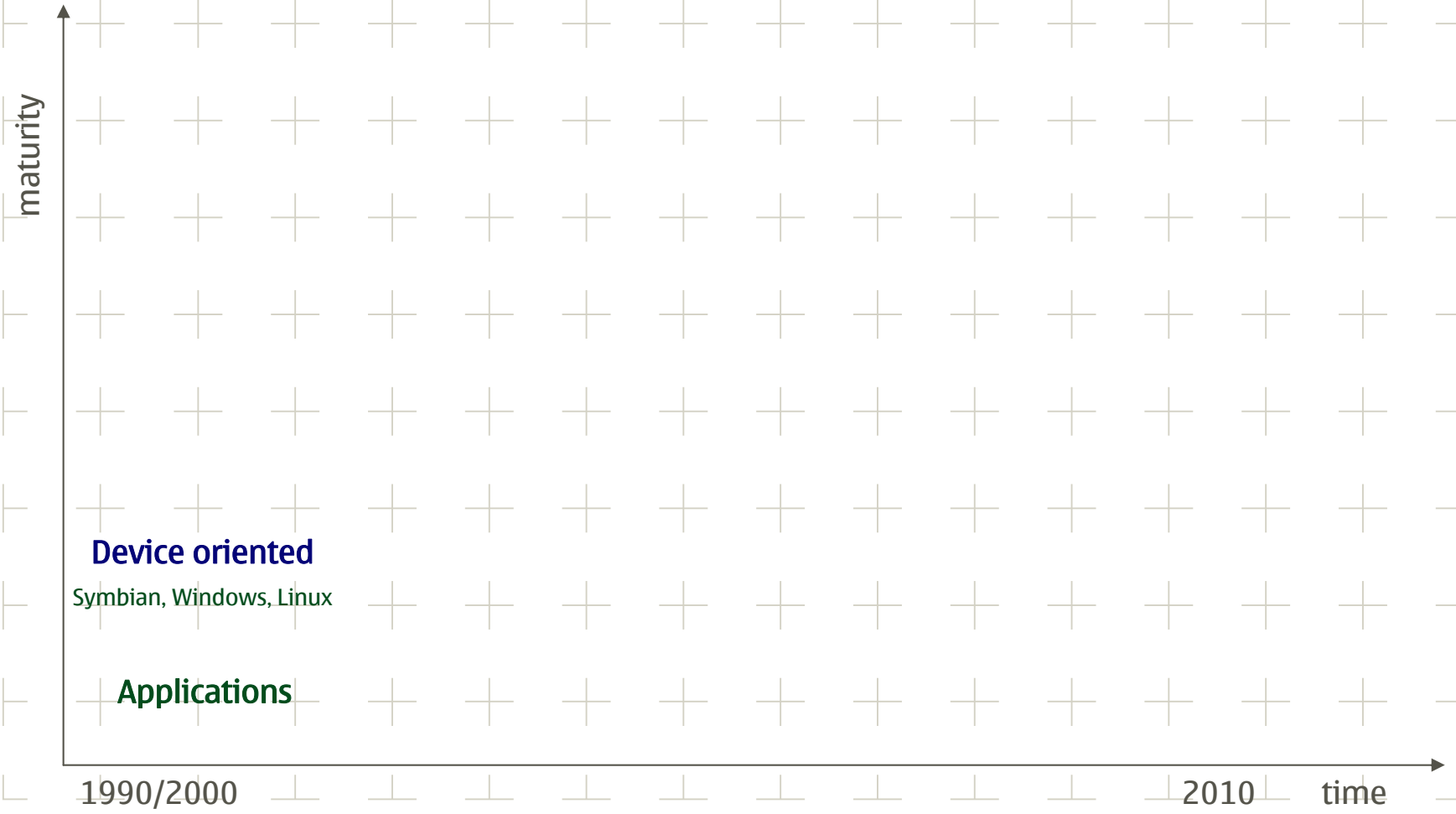
History


Semantic Computation
maps
facebook
blogging
flickr
location
Semantic Web
Web 3.0
tagging
uucp
usenet
Web 2.0
gopher
ftp
www
Web 1.0
1990/2000
2010
time

Technologies - Applications



Technologies - Applications

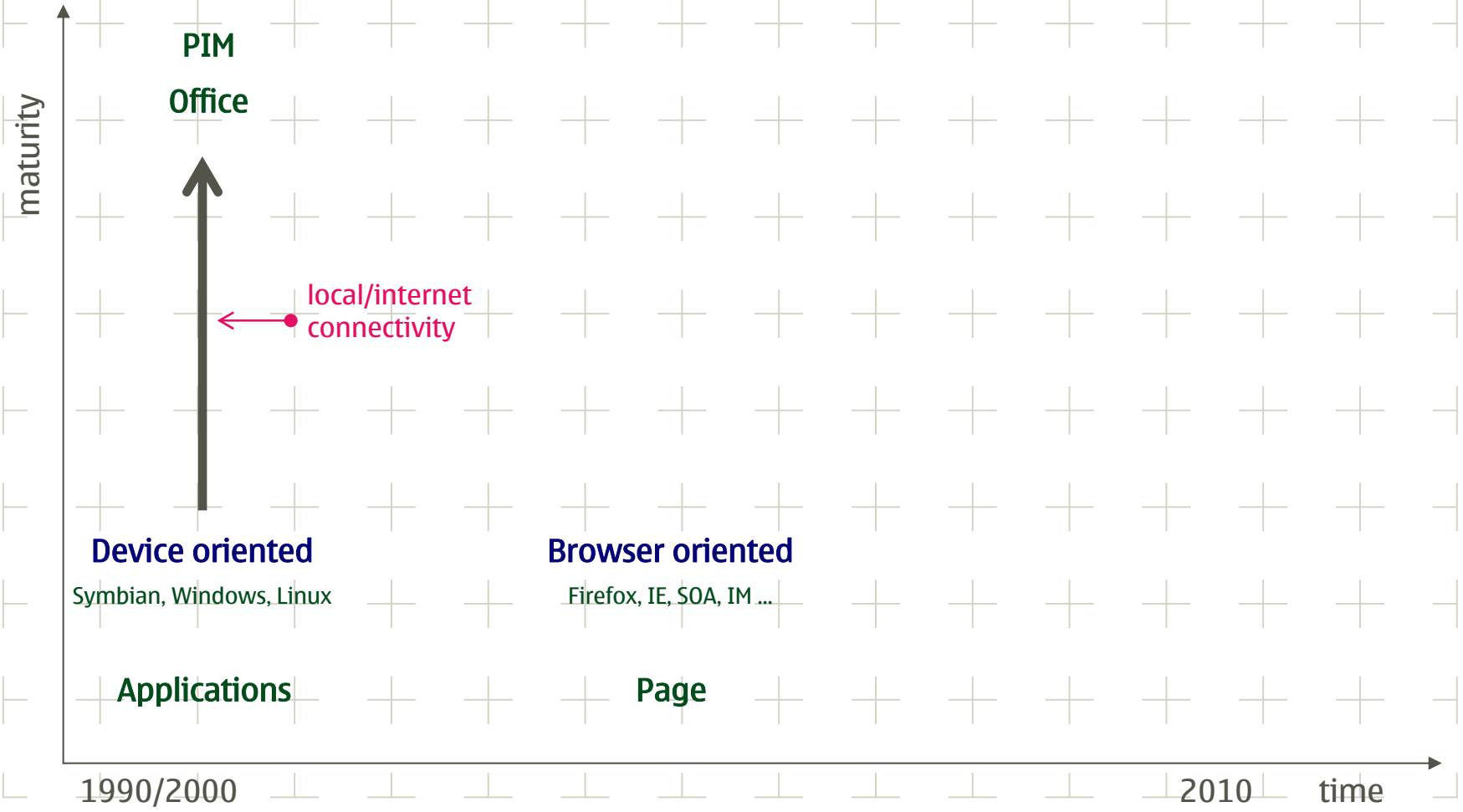


Nokia Research Center

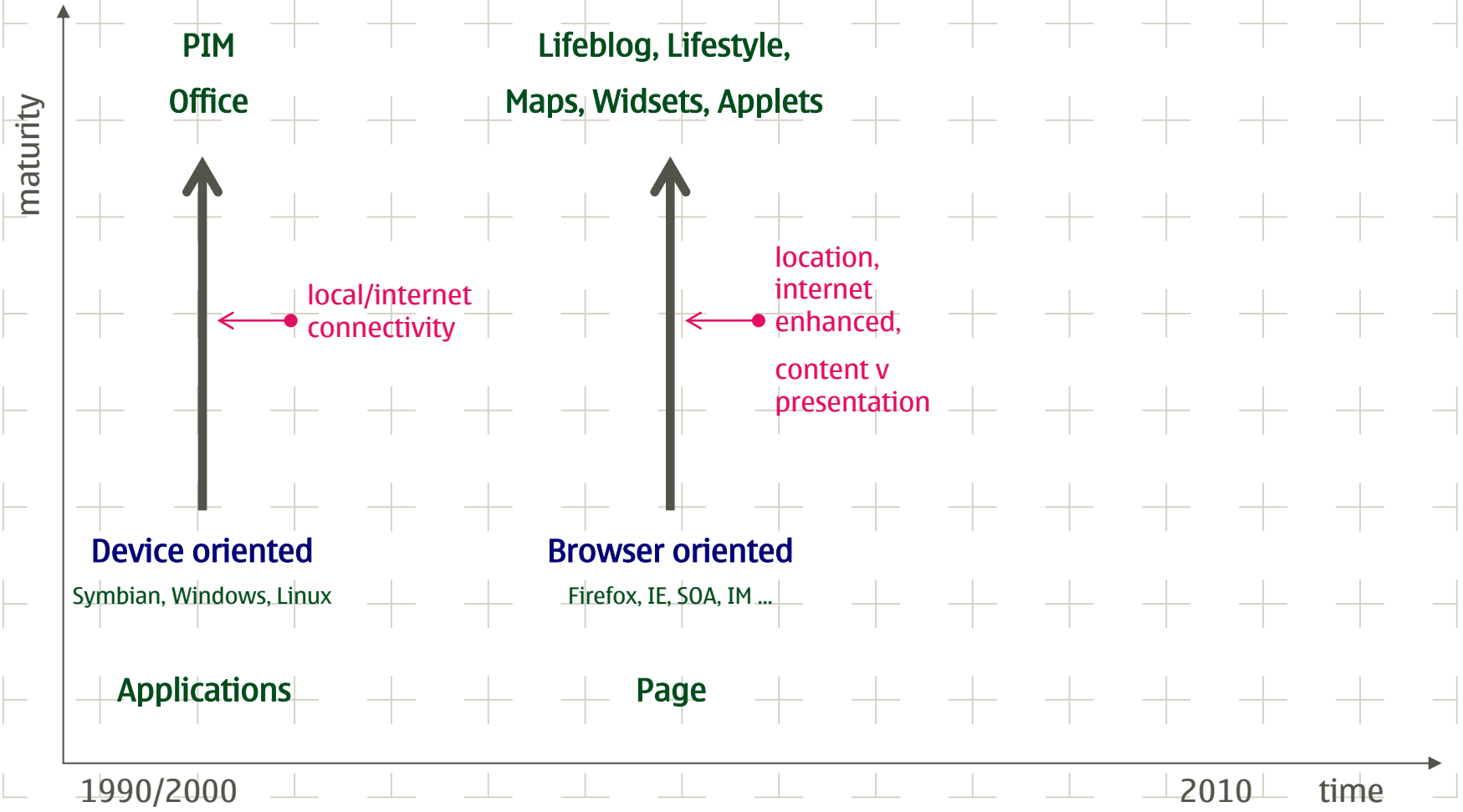
NOKIA

Technologies - Applications

Technologies - Applications



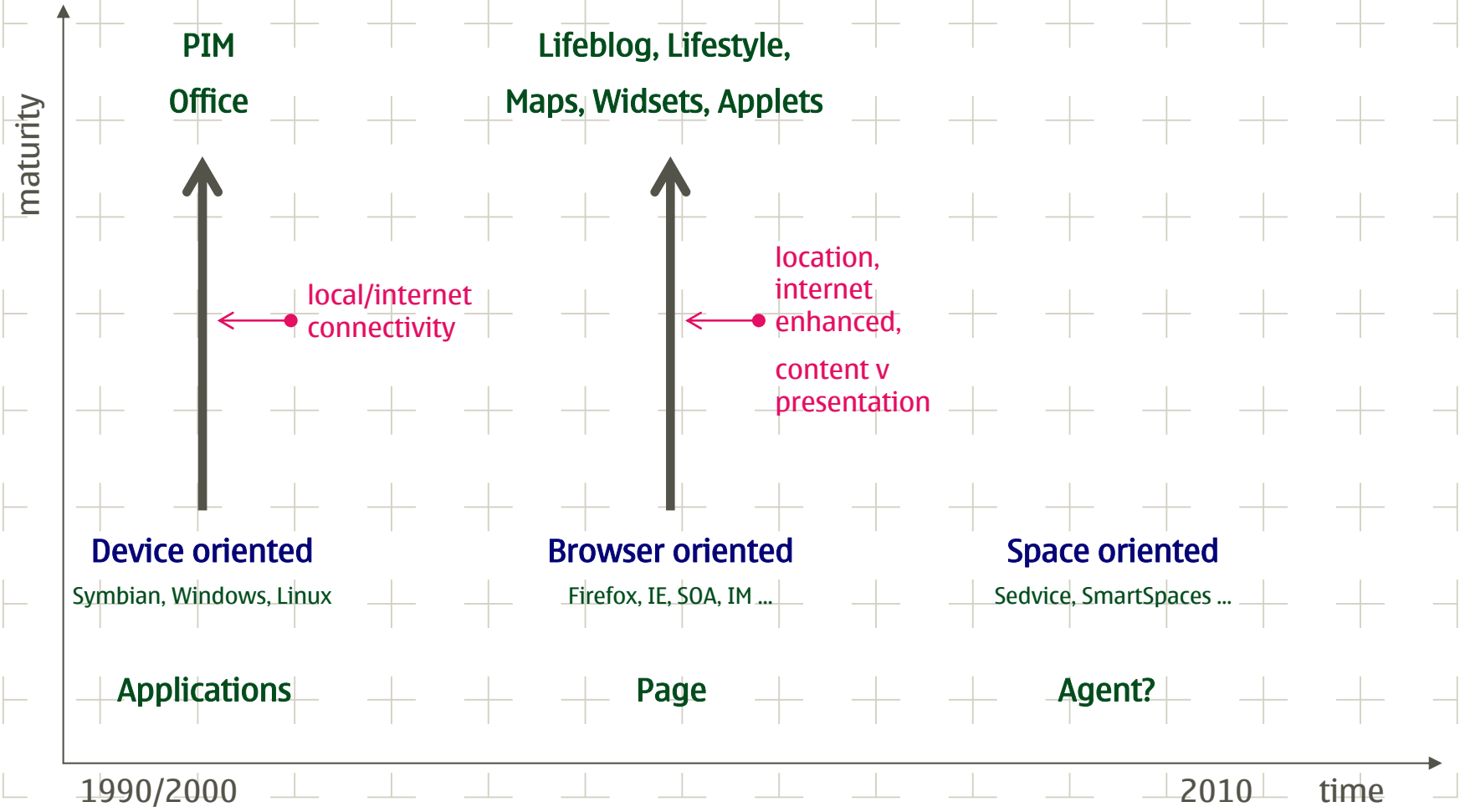
Technologies - Applications



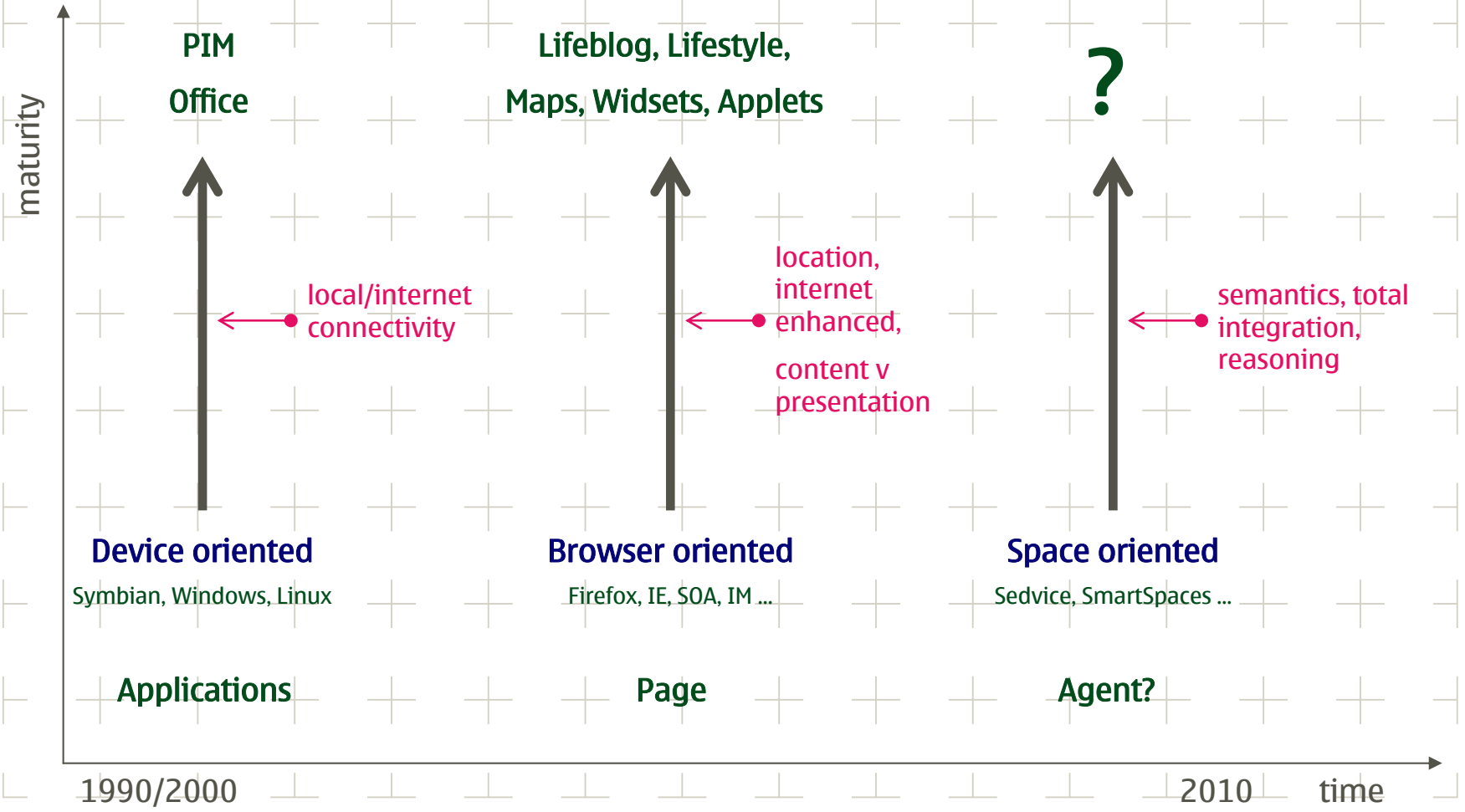
Nokia Research Center

NOKIA

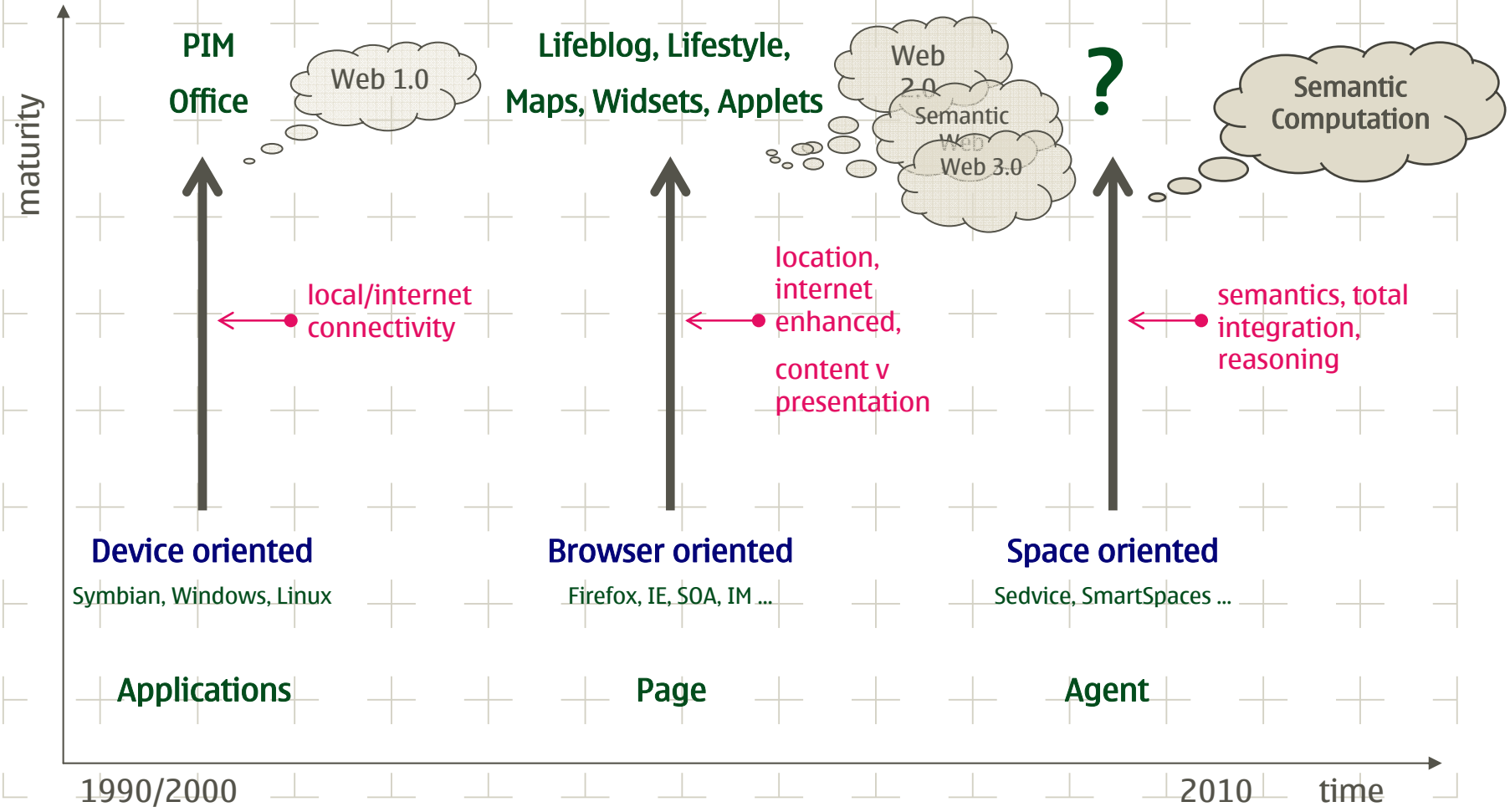
Technologies - Applications



Technologies - Applications



Technologies - Applications



Part Two

Nokia Research Center

NOKIA

Web 1.x, 2.x, 3.x & Semantic Web Characteristics

Webs 1.0, 2.0, 3.0

- content oriented
  - ·news, media
- user publishing
  - ·user generated content
  - ·personal content (gmail, flickr, geotagging etc)
  - ·folksonomies, tagging
- search
  - ·Google, Yahoo

Semantic Web

- information oriented
- classification
- rise of the ontology
  - · strict and structured
  - · enables reasoning, “AI” etc
- global information
- internet of things
- The Giant Global Graph

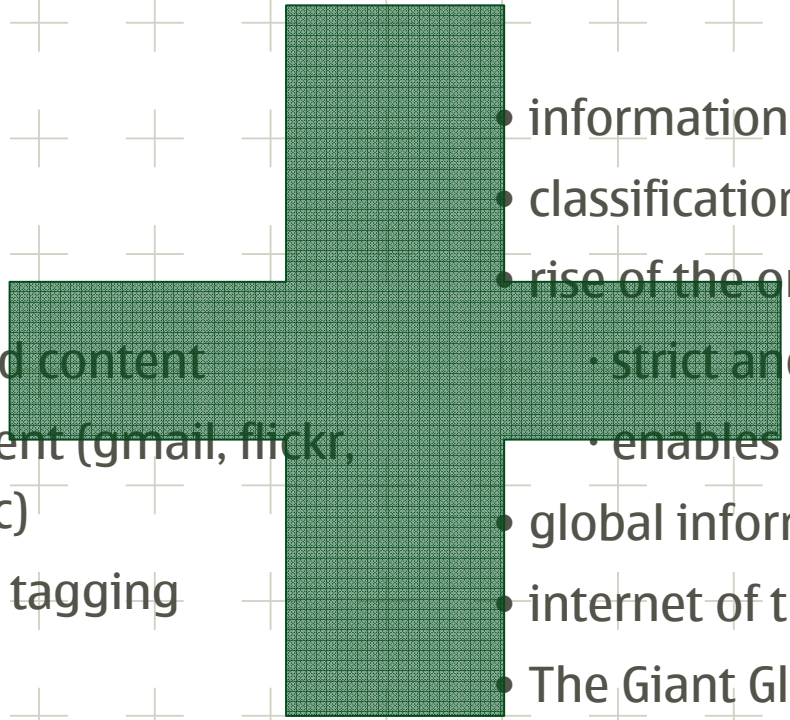
Semantic Computation

Webs 1.0, 2.0

- content oriented
  - ·news, media
- user publishing
  - ·user generated content
  - ·personal content (gmail, flickr, geotagging etc)
  - ·folksonomies, tagging
- search
  - ·Google, Yahoo

Semantic Web

- information oriented
- classification
- rise of the ontology
  - · strict and structured
  - · enables reasoning, “AI” etc
- global information
- internet of things
- The Giant Global Graph



+ a Model of Computation

Semantic Computation – a definition

at least, an attempt at a definition...

"Semantic Computation takes the current Web x.y and Semantic Web concepts and unifies them into a global, ubiquitous computation framework that enables total integration of information in localised, personal contexts..."

Semantic Computation - concepts

Return of the agent

- done before?

Return of the space

- done before?

Semantic Computation - concepts

Return of the agent

- done before?

Return of the space

- done before?

Yes, but...

- it failed...where are the ubiquitous agents today?

Semantic Computation - concepts

Return of the agent

- done before?

Return of the space

- done before?

Yes, but...

- it failed...where are the ubiquitous agents today?

- lack of infrastructure
- lack of ubiquitous computation resources
- lack of representation formats
- lack of classification hierarchies
- lack of standardisation
- lack of global understanding of semantics

Semantic Computation – a definition attempt 2

Context + Reasoning + Agents = Semantic Computation

Semantic Computation – a definition attempt 2

Context + Reasoning + Agents = Semantic Computation

Spaces provide the “closed” (bounded?) environments to compute in.

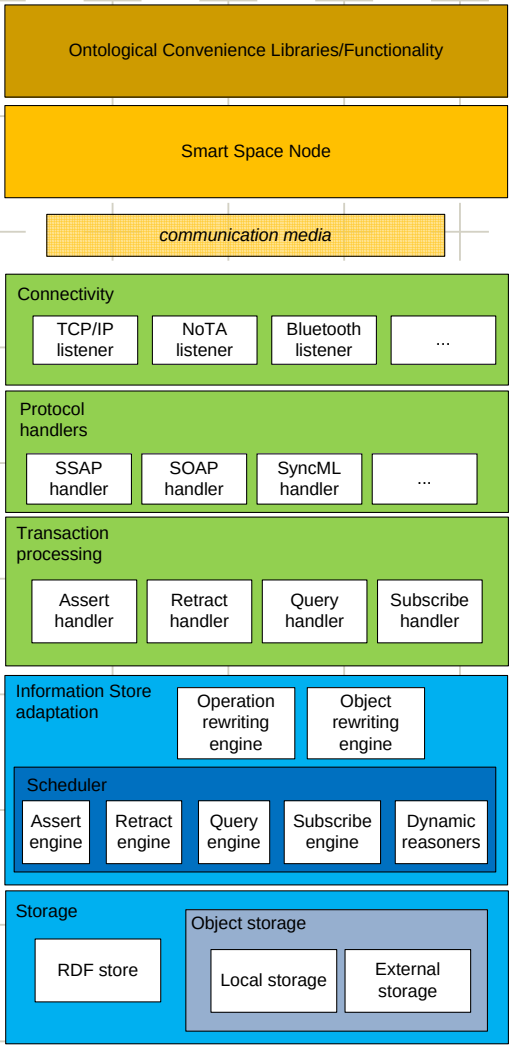
Sedvice-M3

"An environment supporting an space and agent-based model of computation in a Semantic Web based Space providing for integration and interoperability between applications and devices through reasoning mechanisms"

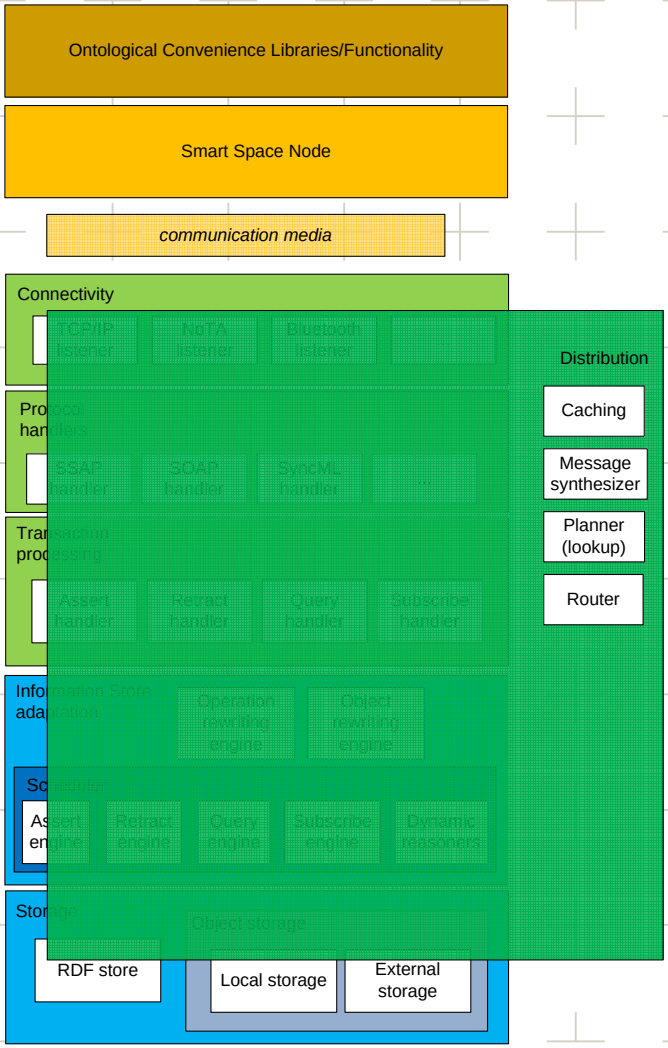
Sedvice-M3 Philosophy

- space-based computing environment
- multiple, individual autonomous spaces
  - local information, reasoning, logics, ontologies etc
  - distributed information
    - distributed deductive closure
    - $D(S_1(Q) \cup S_2(Q)) \neq D(S_1(Q)) \cup D(S_2(Q))$
- information sharing
  - RDF, Semantic Web
  - ontologies, tagging, folksonomies
- applications
  - constructed from agents
  - autonomous, anonymous, distributed, mobile
  - control-flow through ontological means
- 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!
  - free logics
  - non-monotonic
- semantics of information, belief and truth maintenance responsibility of the reader (agent/actor)
  - everything is information
  - everything is first-order
- first-order policy, security, belief and trust structures

Sedvice-M3 Architecture



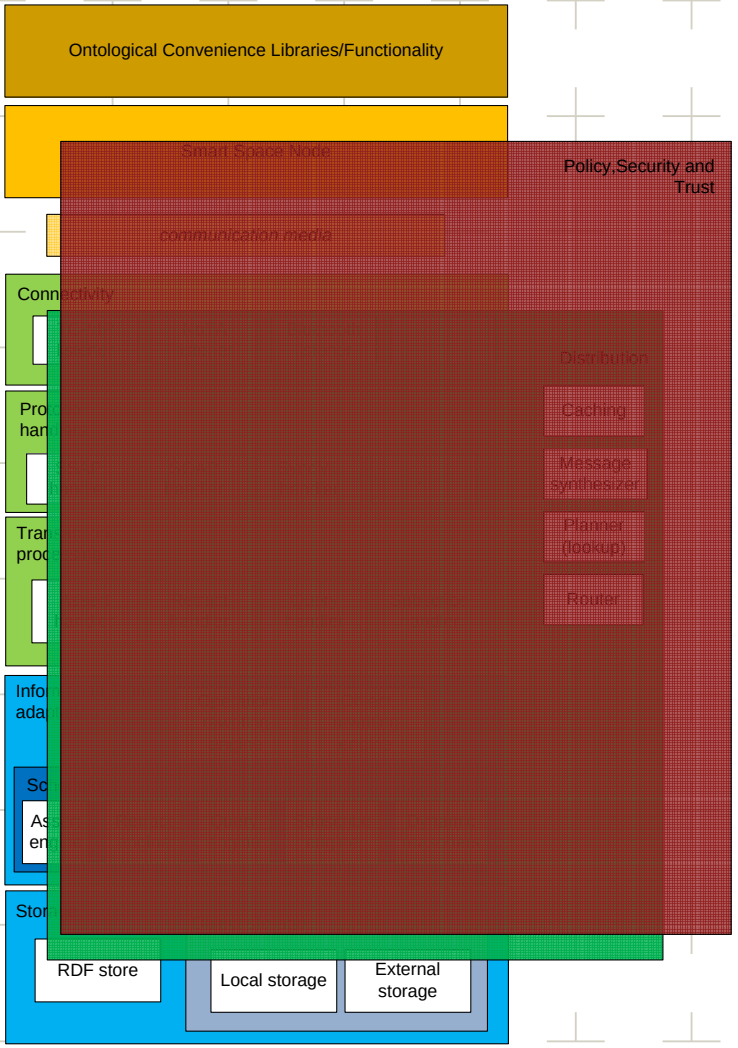
Sedvice-M3 Architecture



Nokia Research Center

NOKIA

Sedvice-M3 Architecture



Nokia Research Center

NOKIA

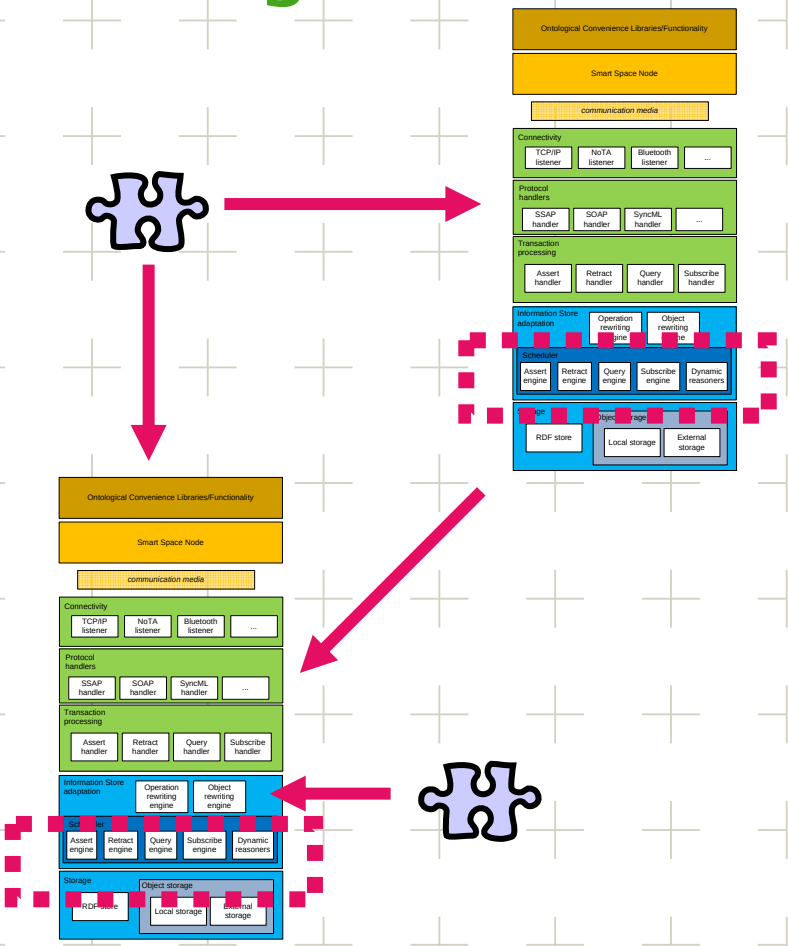
Sedvice – Interal vs External Reasoning

External Reasoning

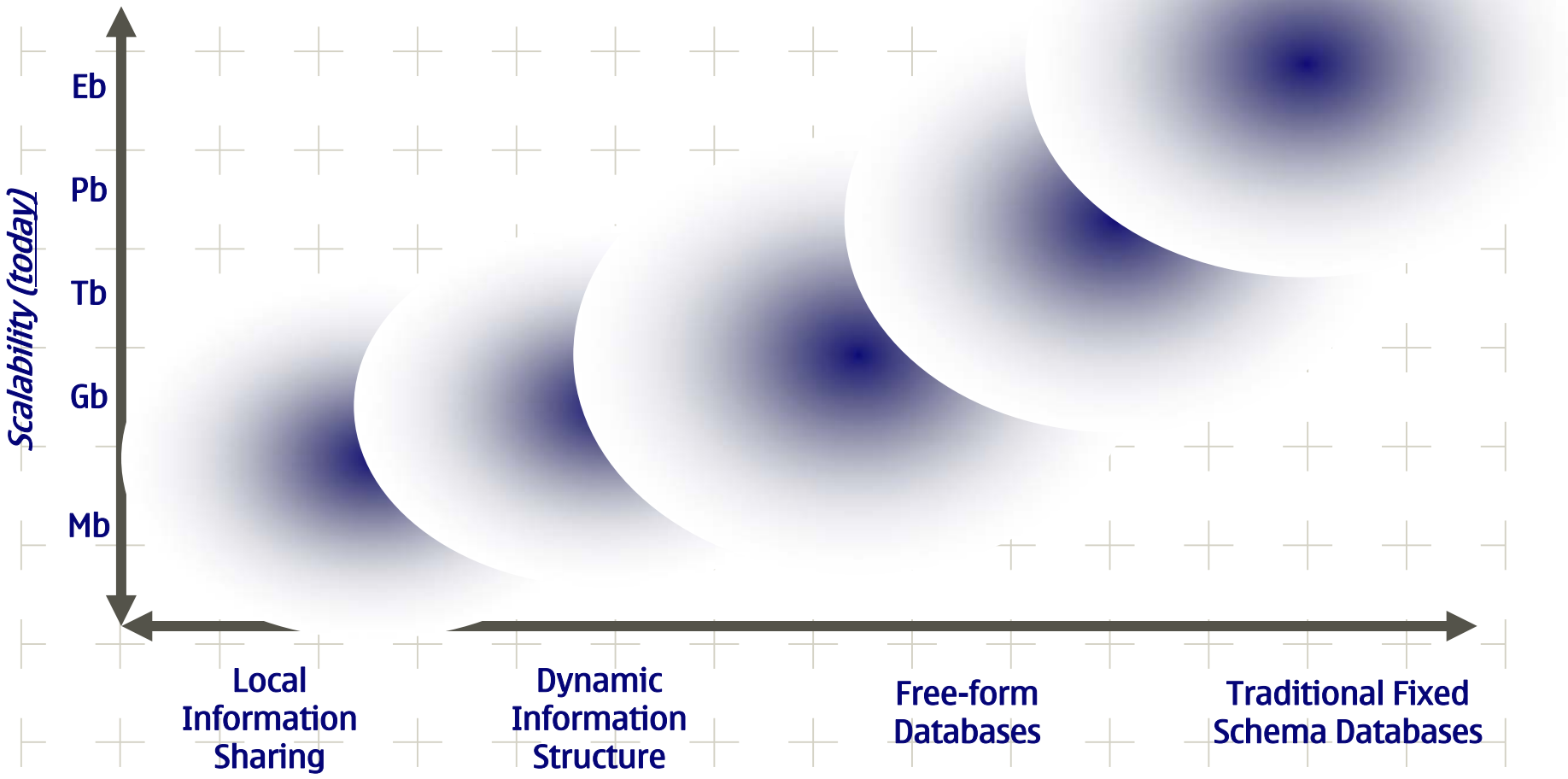
- by agent
- by shared space

Intenal Reasoning

- statically
  - RDF++, RDF#?
  - deductive closure calculation
- internal agents
  - restricted execution environment



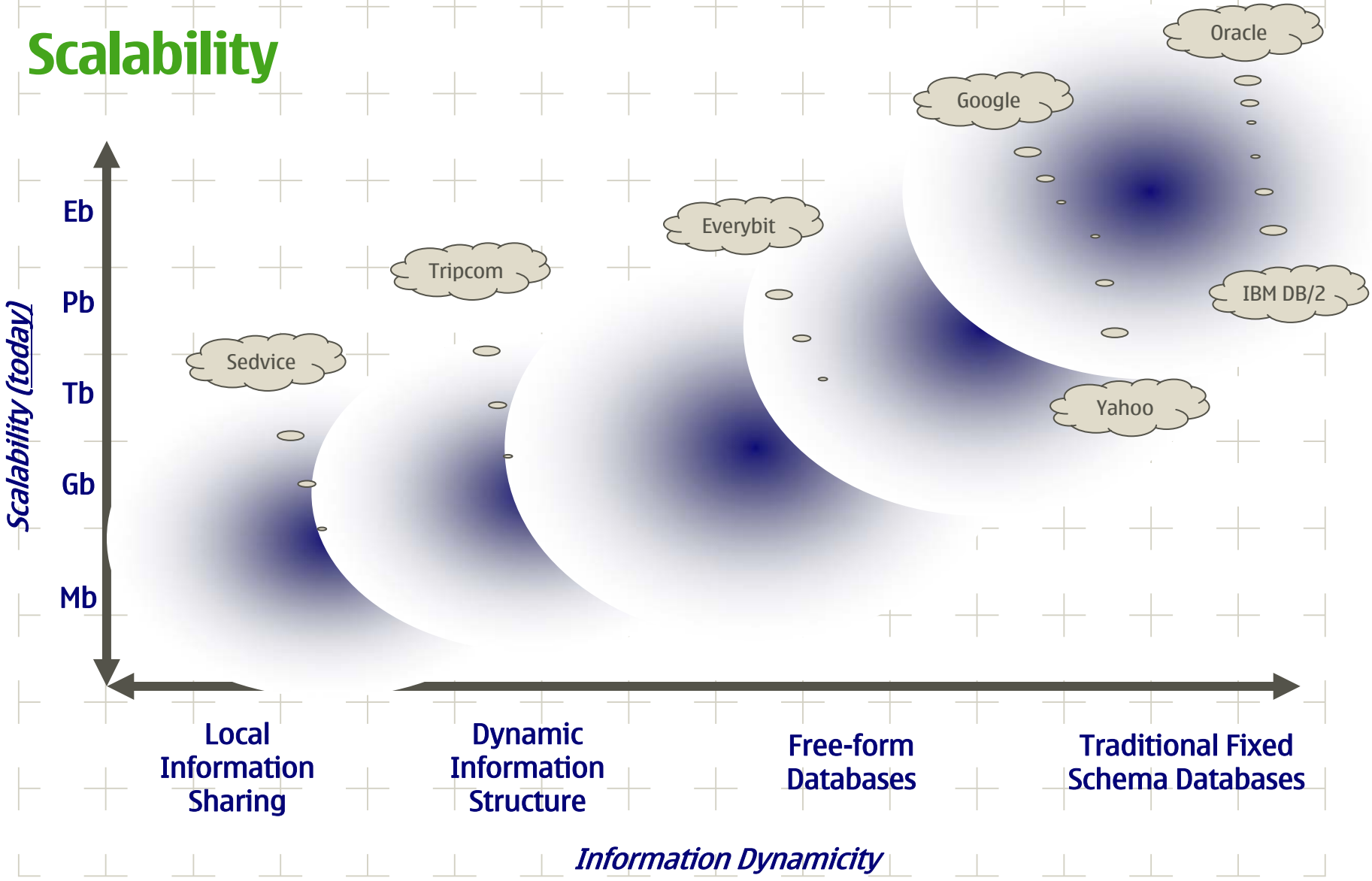
Scalability



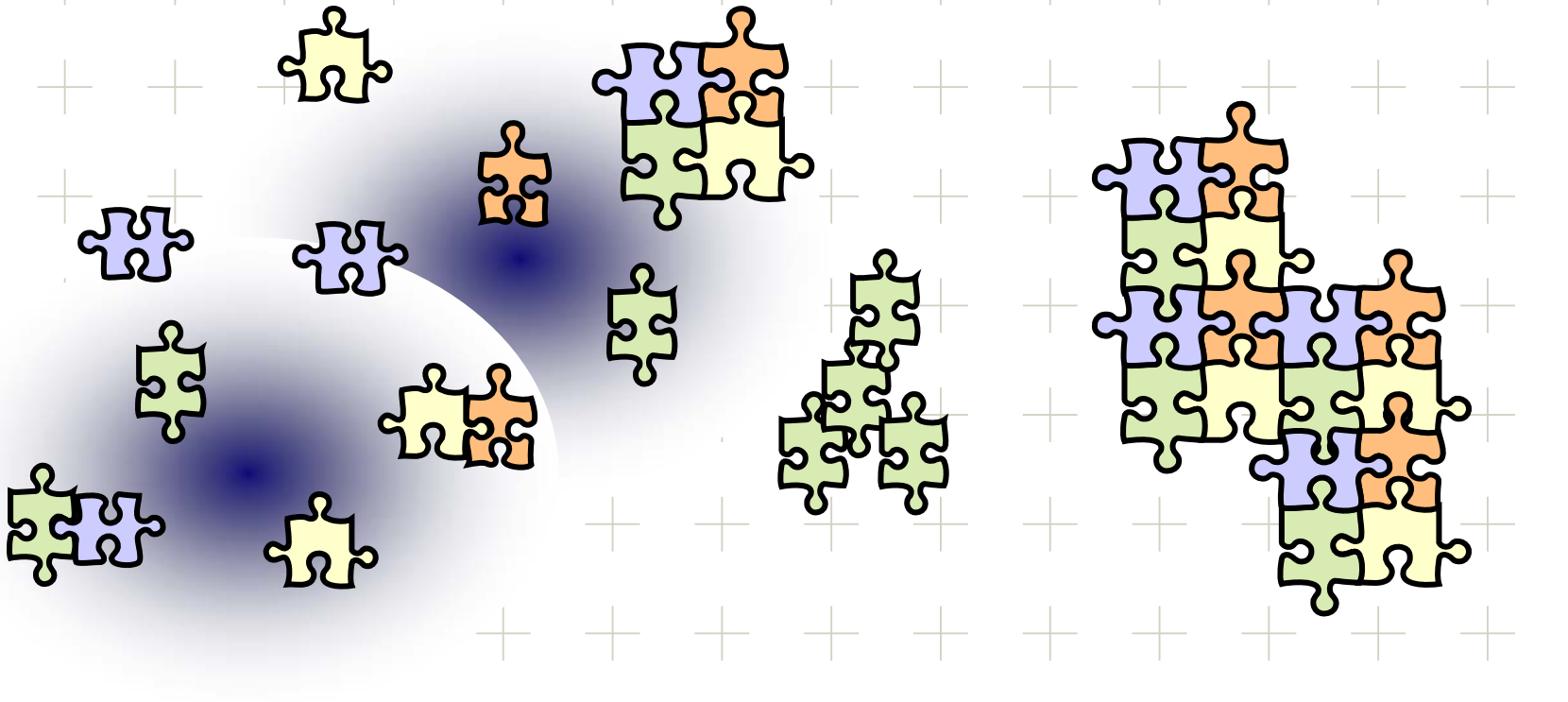
Nokia Research Center

NOKIA

Scalability

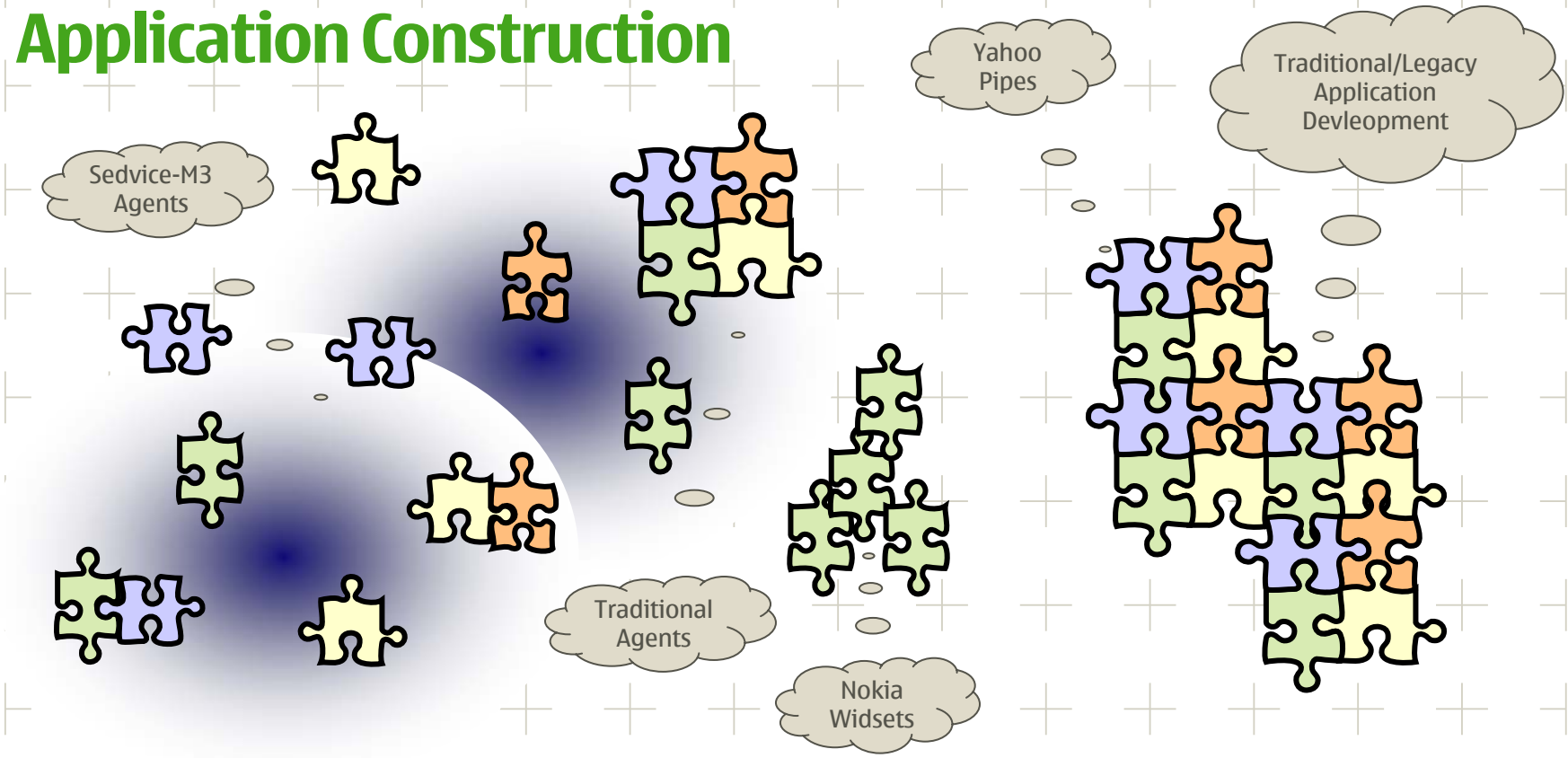


Application Construction



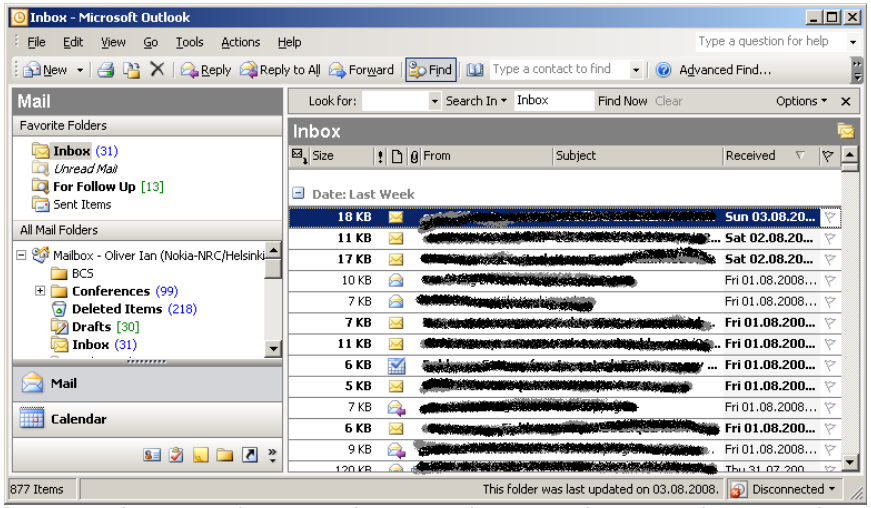
Application Construction

Application Construction



Application Construction

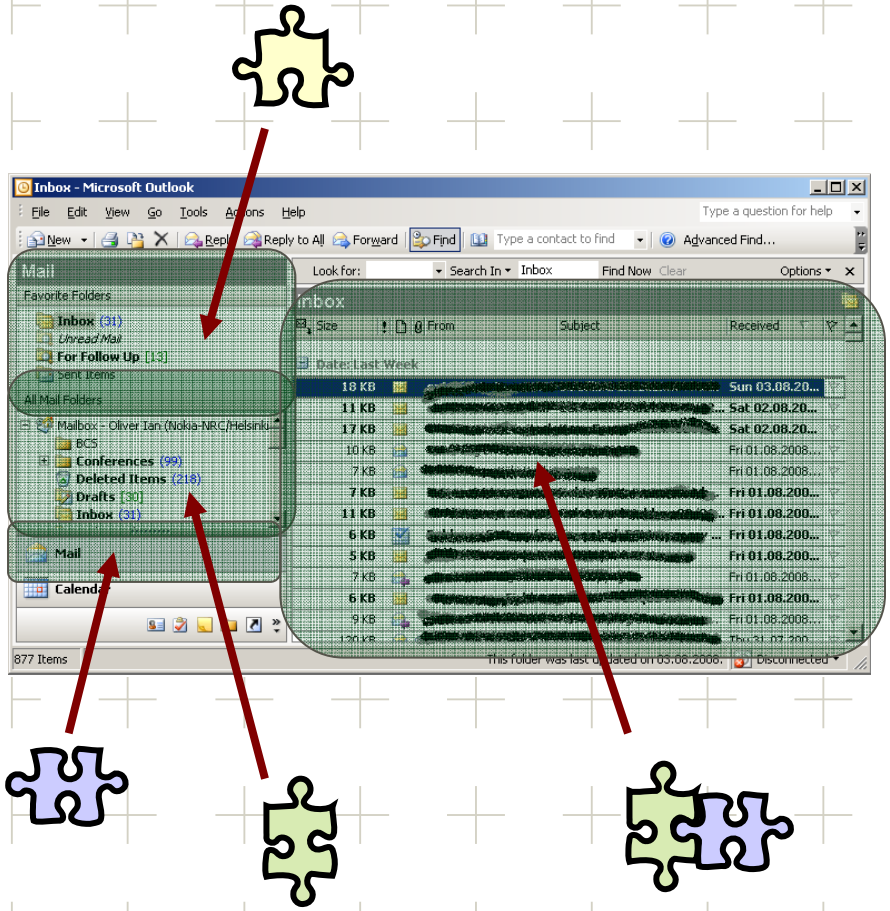
Applications



Traditional applications

- monolithic
- single purpose
- difficult to expand and enhance
- fixed focus
- integration with other applications
  - impossible in an ad hoc manner

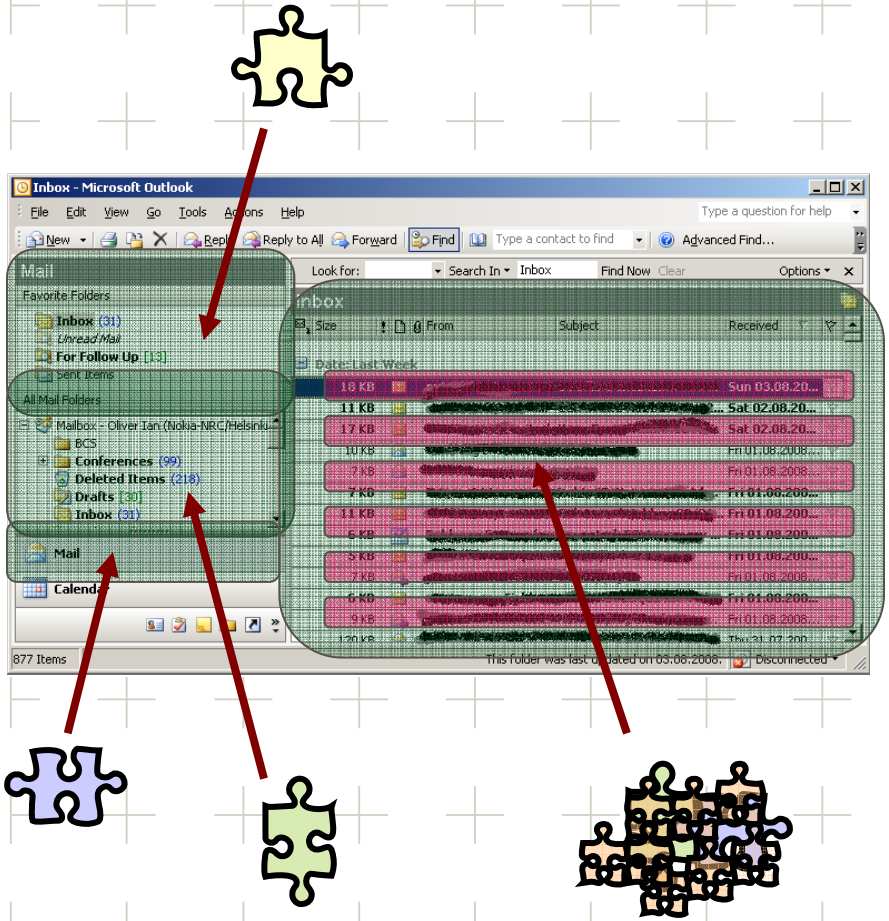
Applications



Semantic Computation based applications

- appearance decided by user
- UI functionality
- individual agent-based parts
- information gathered from numerous sources
- reasoning about information
  - eg: weather reports as email
  - friends = contacts
  - locations = contacts
  - etc
- functionality is emergent

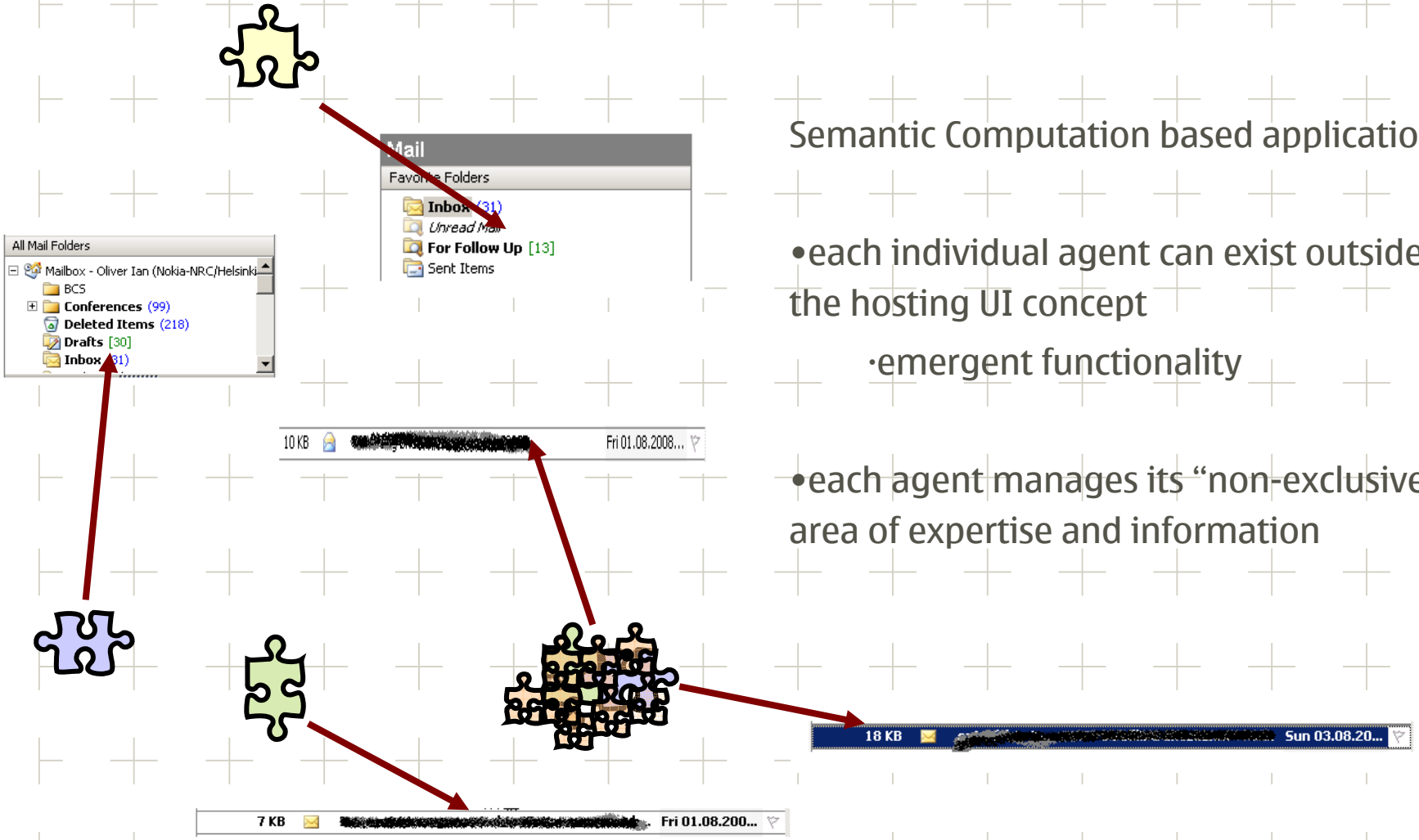
Applications



Semantic Computation based applications

- granularity can be extremely fine

Applications



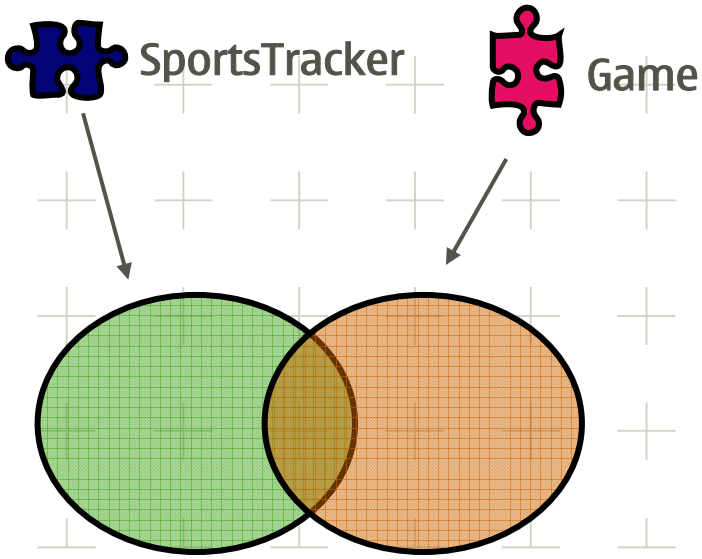
Semantic Computation based applications

- each individual agent can exist outside of the hosting UI concept
  - emergent functionality

- each agent manages its "non-exclusive" area of expertise and information

Examples

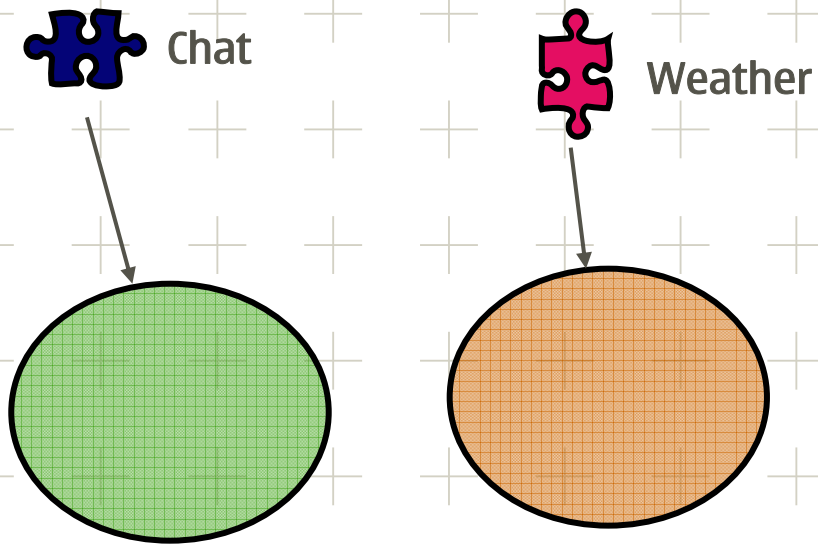
- Simple Application Interaction



- Nokia Sports Tracker
  - writes current exercise information to a space
- Game
  - subscribes for whether user has exercised recently
  - awards extra (or less!) lives depending upon the above
- *Jukka Honkola, Hannu Laine et al...*

Examples

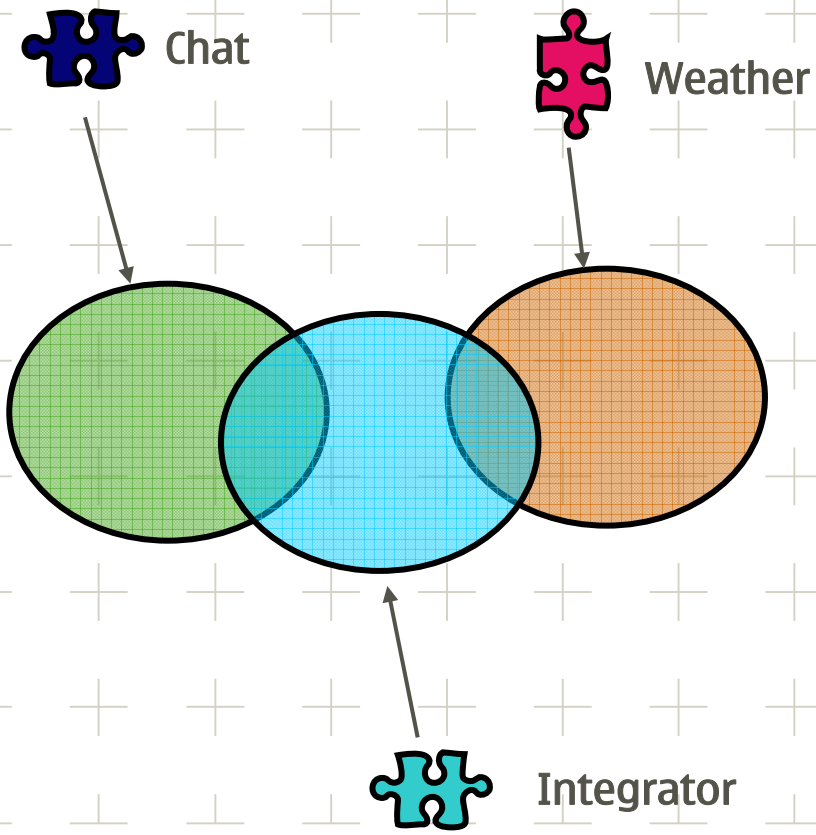
- Complex Interaction



- Chat
  - "traditional" IM-style chat
- Weather Feed
  - obtains weather reports for a set of given cities
  - ...

Examples

- Complex Interaction



- Chat
  - "traditional" IM-style chat
- Weather Feed
  - obtains weather reports for a set of given cities
- Integrator
  - Monitors chat for city names and injects weather reports into chat as conversation messages (multiple typing!)

Chat...


Message writer


Message reader


Conversation joiner


Conversation watcher

- Atomic Functionalty = Agent

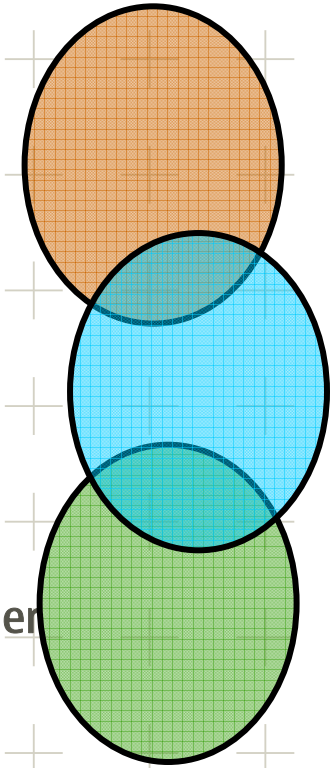
Chat...


Message writer


Message reader

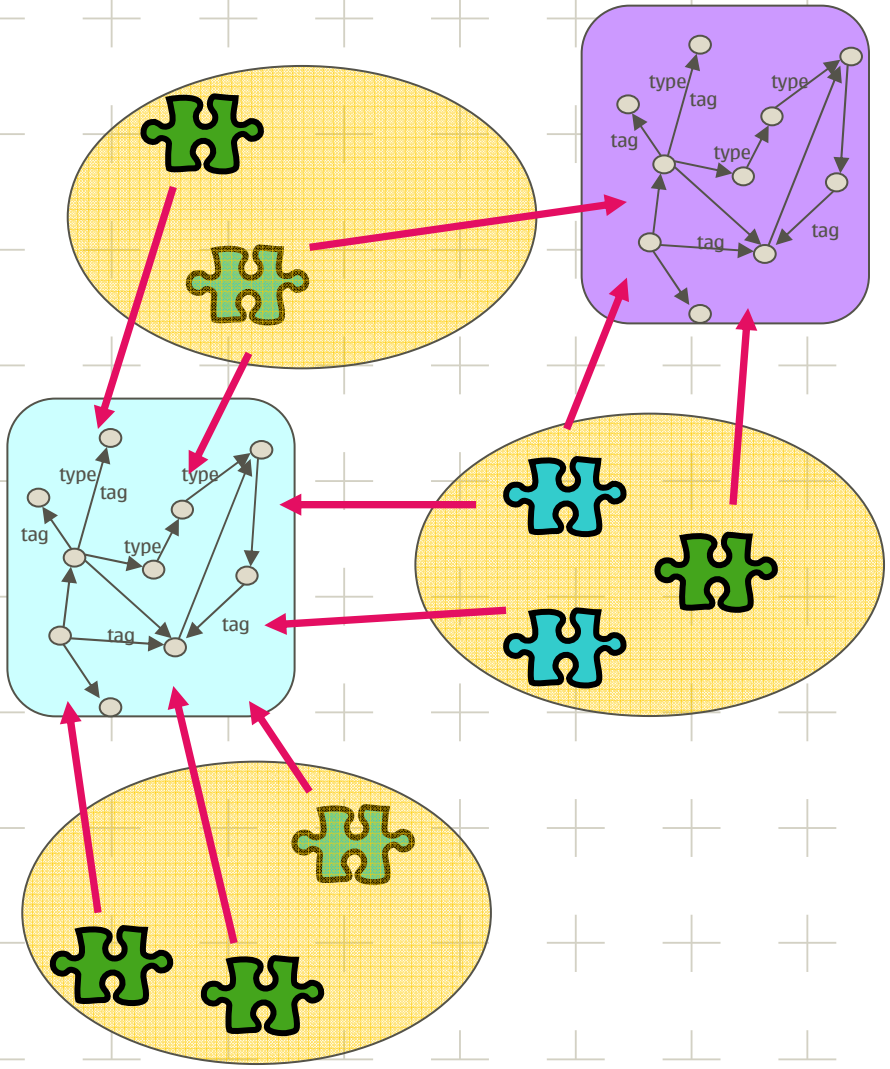

Conversation joiner


Conversation watcher



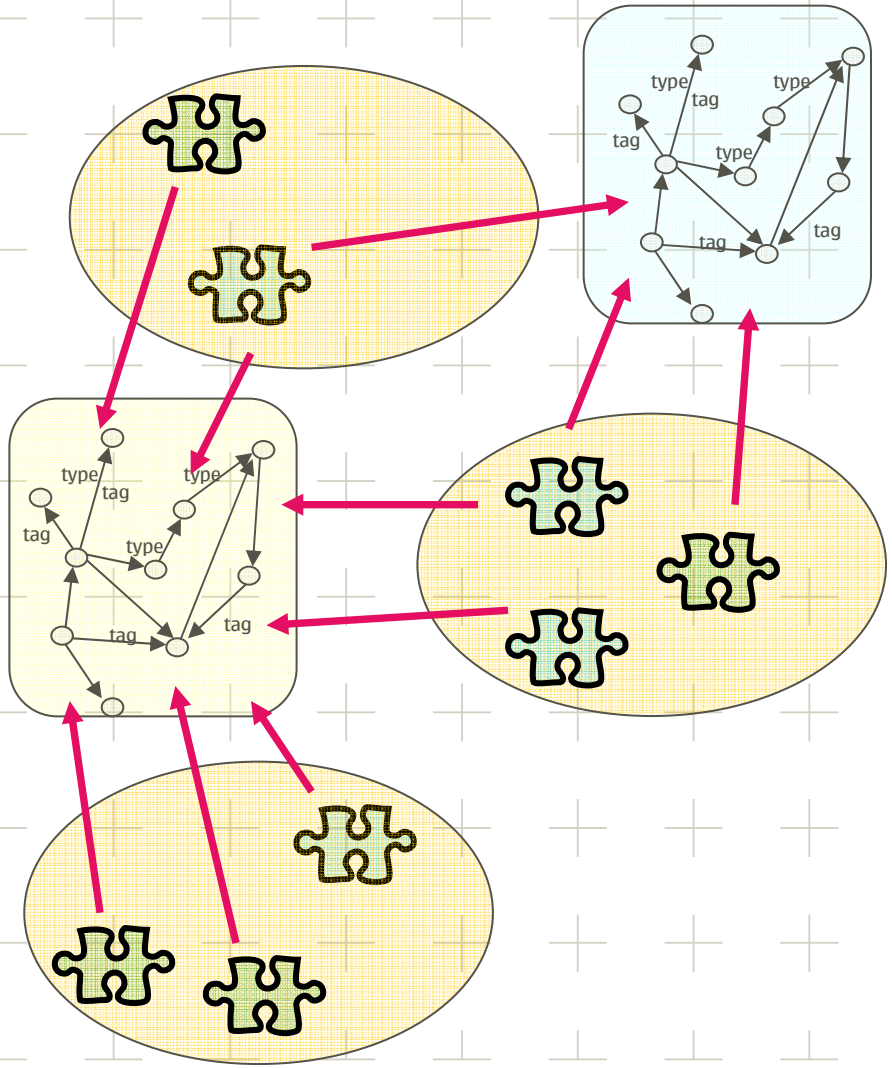
- Atomic Functionalty = Agent
- Each agent is then responsible for a certain subset of the “ontology(ies)” used in the “application”

Chat...



- Agents now may be distributed across multiple devices and communicate through various, related spaces
- A certain number of agents are required to fulfill the application
  - exact number depends upon the situation
  - too many/too little = application degeneration

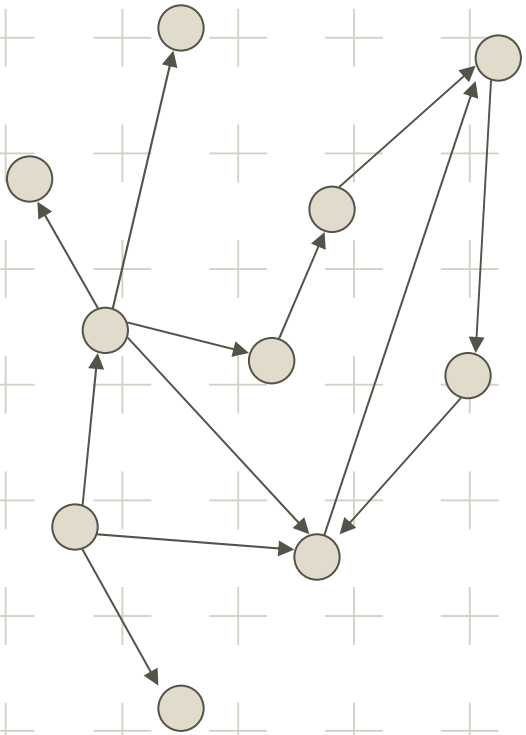
Possibilities...



- Personal
- Home
- CityScape
  - ·tourism
  - ·local context
  - ·airport
    - ·tracking
- Work

+ any combination thereof...
  - ·feeder information
  - ·ad hoc social blogging
  - ·???

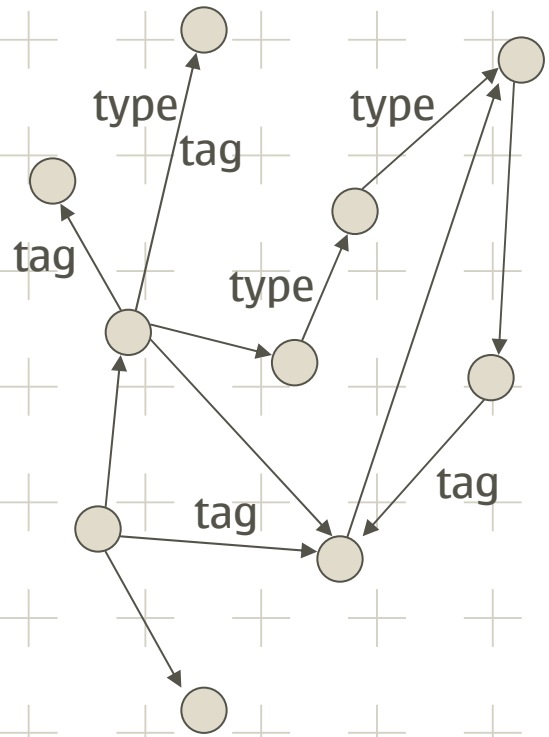
Theoretical Underpinnings - Graph



Information is a (directed) graph

- RDF + Reasoning
- RDF++ / Wilbur
- Everything is first-class

Theoretical Underpinnings - Graph

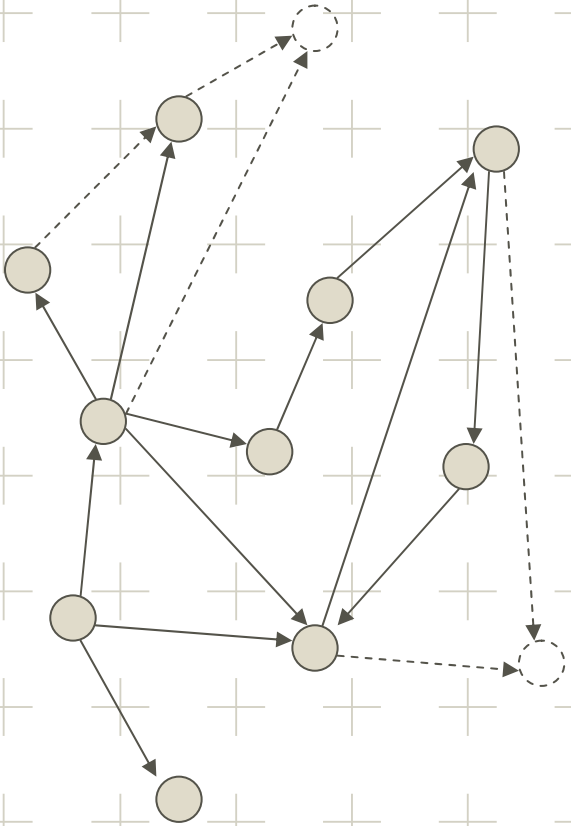


Within a graph information may be typed according to some ontology

or, tagged according to some folksonomy

or, both

Theoretical Underpinnings – Graph Structures

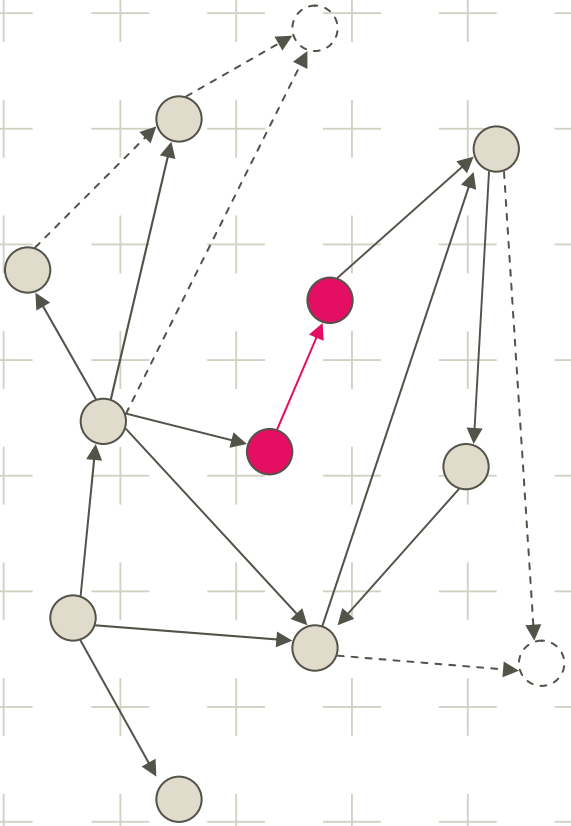


RDF Graph = mathematical graph + additional constraints and deduction

- $\Gamma, \forall x \in X, X \subseteq Y, Y \subseteq Z \Rightarrow X \subseteq Z \therefore \forall x \in Z$
- these rules can be modified by the space's logic

- Deduction is performed at query-time, ie: dyamically
  - some spaces might perform deduction at insert-time

Theoretical Underpinnings – Graph Structures

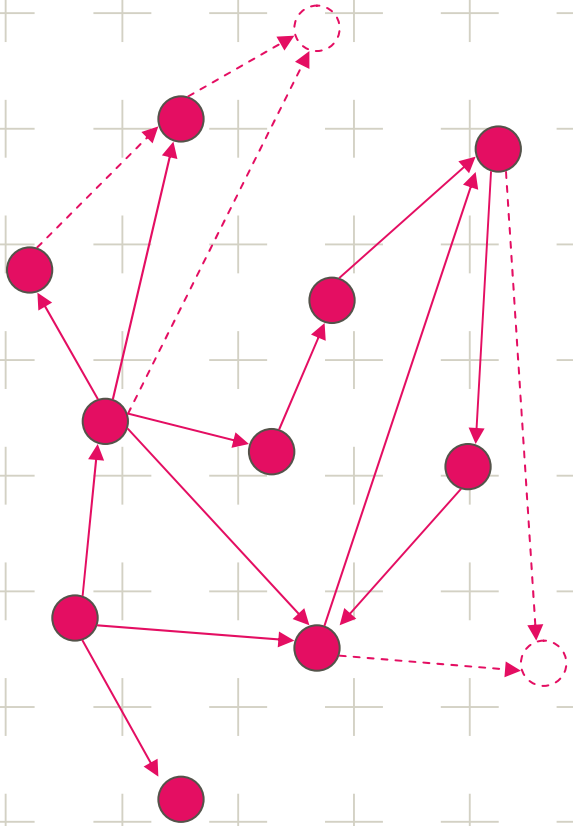


Two extremes:

RDF Graph

RDF Triple

Theoretical Underpinnings – Graph Structures

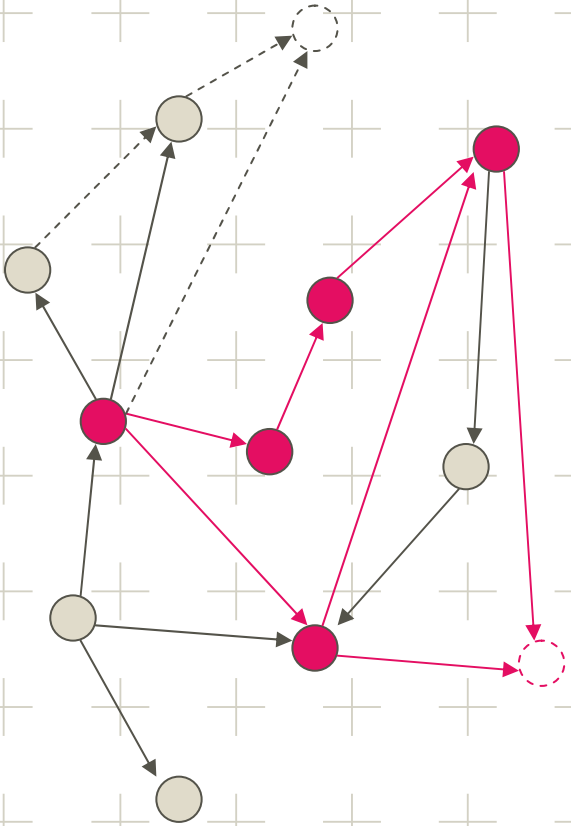


Two extremes:

RDF Graph

RDF Triple

Theoretical Underpinnings – Graph Structures



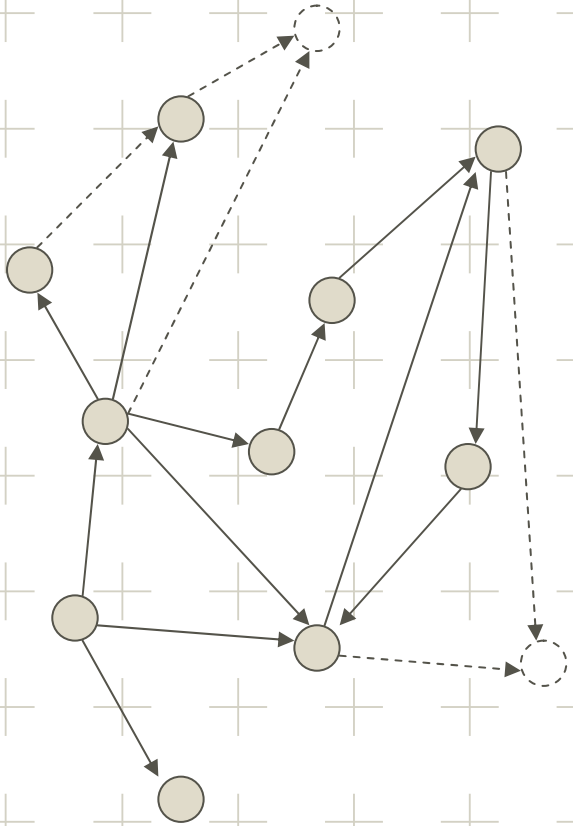
Finer grained mechanisms necessary

RDF Graph

Molecule
Subgraph
Scope

RDF Triple

Theoretical Underpinnings – Scopes & Reflection

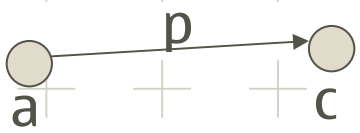


First-order characteristics of scope

- Reflection
- Scopes can be composed
- Scopes require additional operators (other than graph traversal)
  - union
  - intersection
  - guards/pre-conditions
  - etc

- RDF as its own programming language as well as representation?

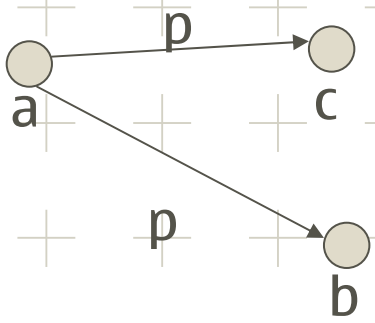
Theoretical Underpinnings – Operations



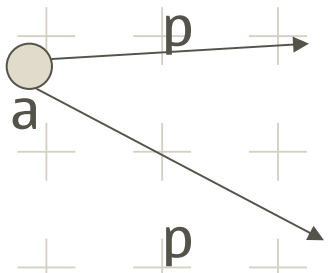
Four basic operations

- graph insertion

insert{(a,p,b)}



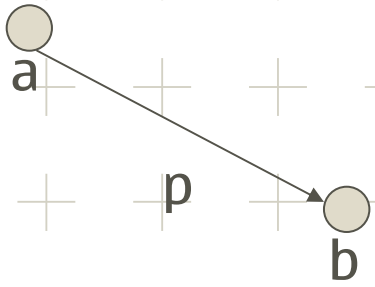
Theoretical Underpinnings – Operations



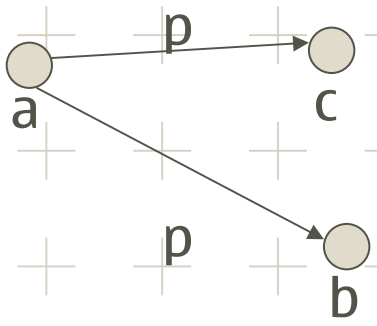
Four basic operations

- graph insertion
- graph retraction

retract{(a,p,c)}



Theoretical Underpinnings – Operations



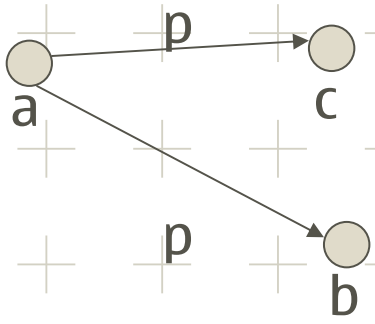
query(a.p)

{ c b }

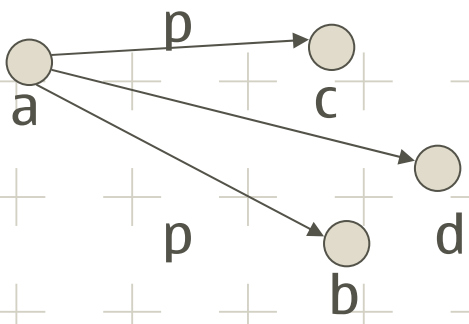
Four basic operations

- graph insertion
- graph retraction
- query
  - synchronous
  - WQL, SPARQL, whatever…

Theoretical Underpinnings – Operations



subscribe(a.p)
insert{(a,p,d)}

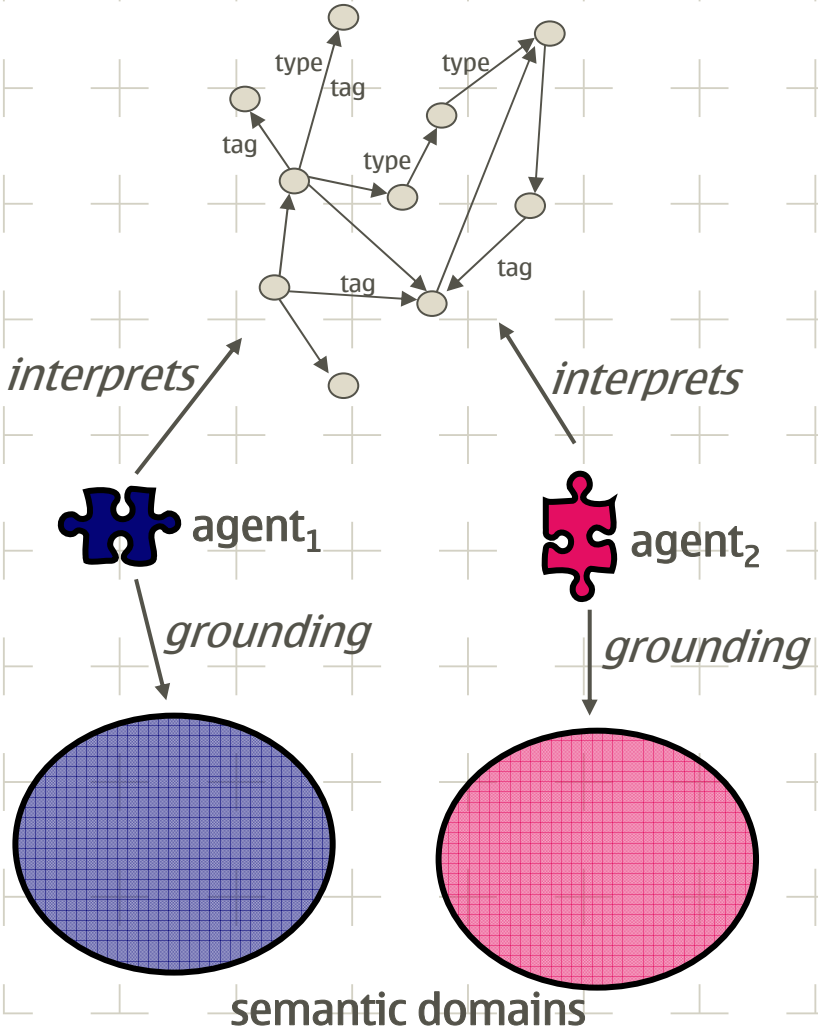


Four basic operations

- graph insertion
- graph retraction
- query
  - ·synchronous
  - ·WQL, SPARQL, whatever…
- subscription
  - ·persistent query

returns: { c b d }

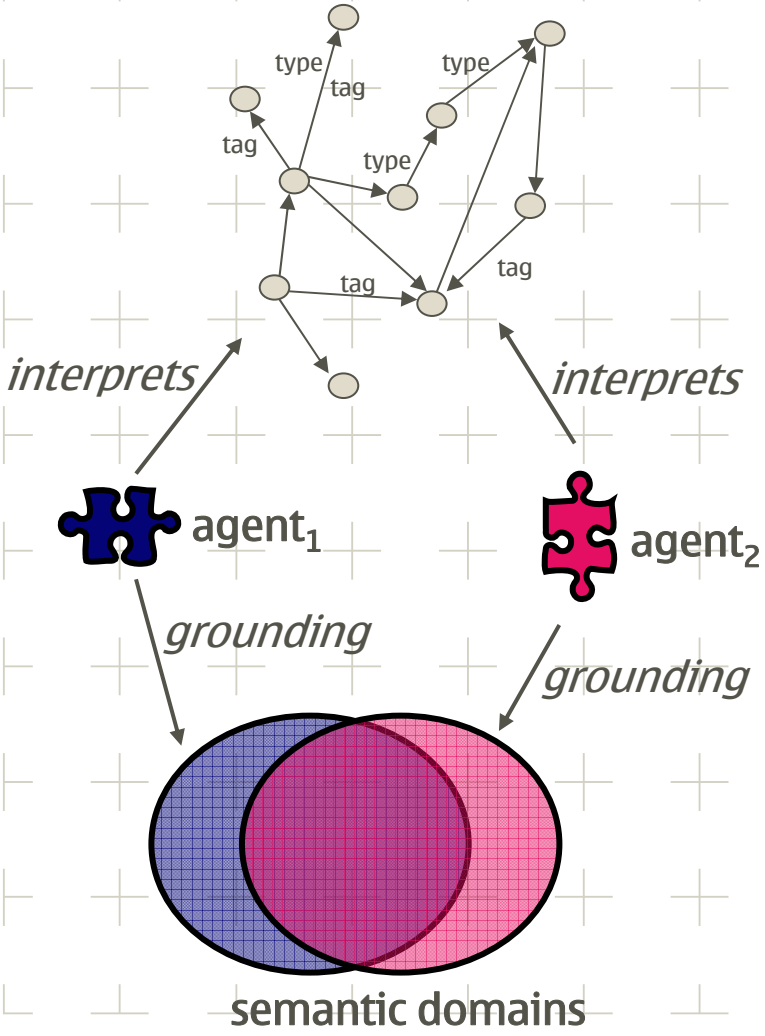
Theoretical Underpinnings – Intentional Semantics



Interpretation to some semantic grounding is made on a per agent basis

ontologies and folksonomies provide *assistance only...*

Theoretical Underpinnings - Intentional Semantics



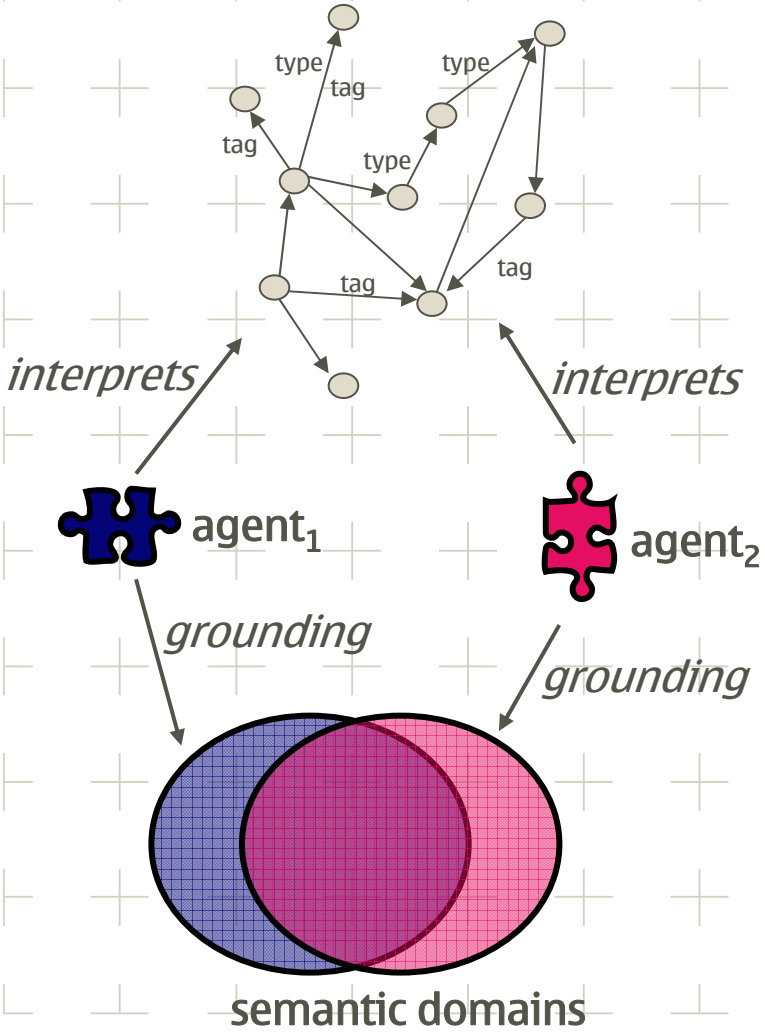
Interpretation to some semantic grounding is made on a per agent basis

ontologies and folksonomies provide *assistance only*...

...which *helps* in ensuring that a common interpretation is made

- at least a common enough interpretation

Theoretical Underpinnings - Intentional Semantics

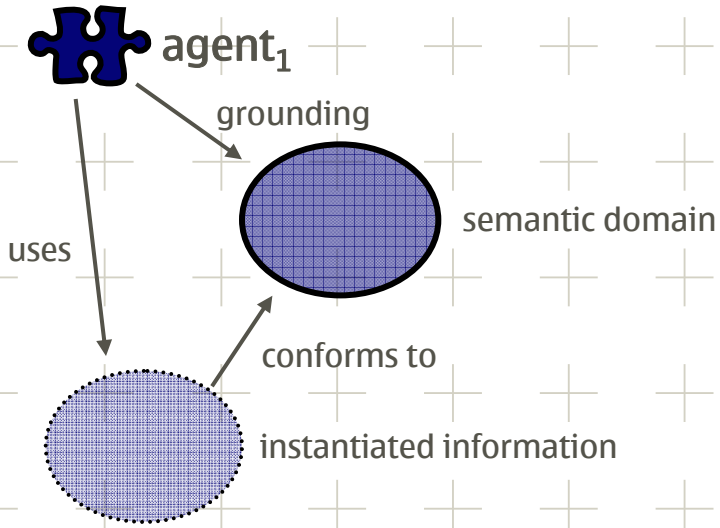


Semantics is intentional rather than fixed.

The agent writing a given piece of information provides meta-information (type, tag, other properties, relationships etc) to indicate its intention how that piece of information should be interpreted

cf: duck-typing, mixins, multiple-inheritance, undecidablity, description logic decision procedures etc...

Theoretical Underpinnings – Agent Interoperability

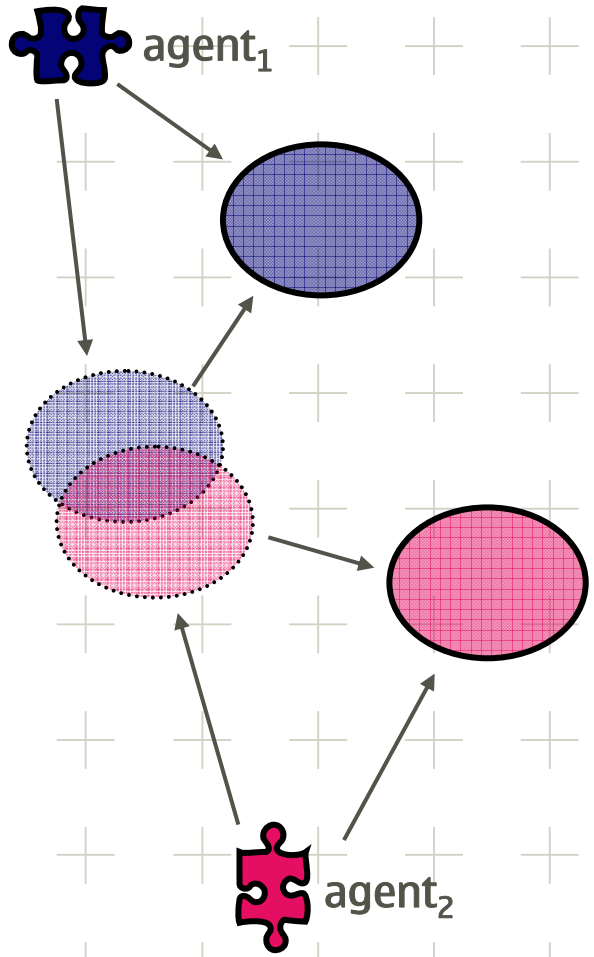


Agents operate over a given subset of information

- deliniated by ontology, tagging and the semantic area of that agent

- the agent interprets that information according to its semantic grounding

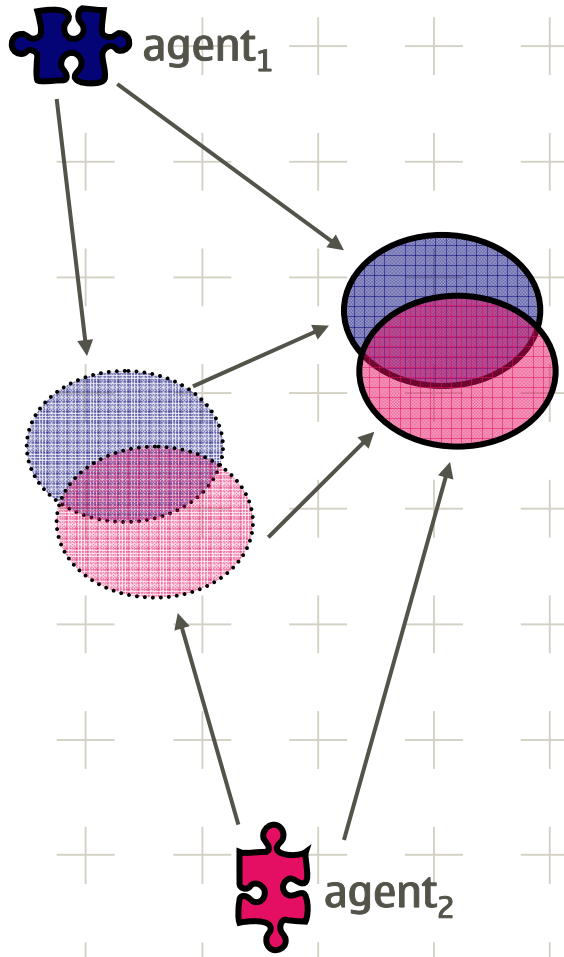
Theoretical Underpinnings – Agent Interoperability



Two agents "communicate" if there is intersection between the information they are using

- here lies a problem
  - there two agents might interpret the information in completely different ways
  - chaos and nonsense might result

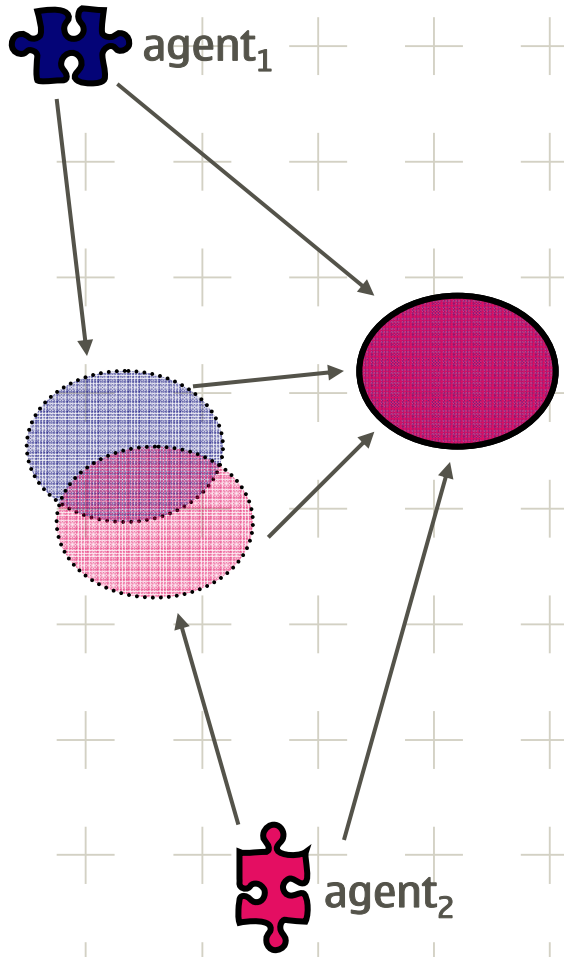
Theoretical Underpinnings – Agent Interoperability



Two agents "communicate" if there is intersection between the information they are using

- sensible communication only results if the semantic domains of the agents are aligned sufficiently
  - we do not have good definitions nor metrics to define "sufficient enough"
  - standardisation....

Theoretical Underpinnings – Agent Interoperability

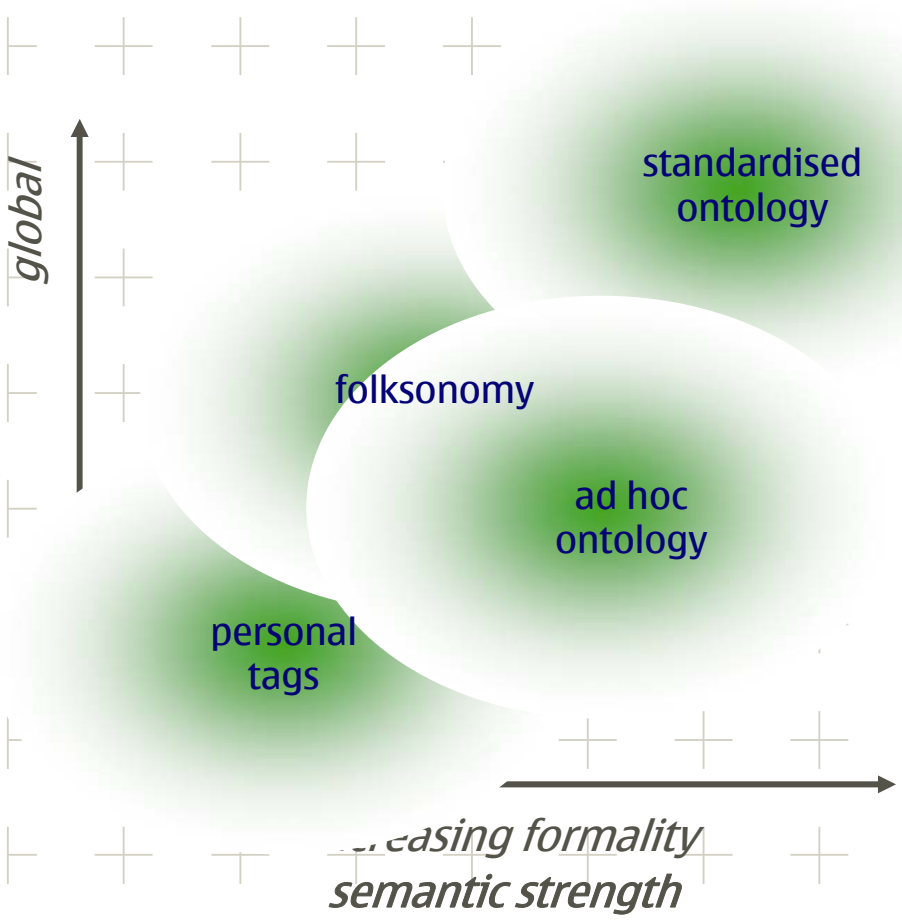


Two agents "communicate" if there is intersection between the information they are using

- harmonious communication and understand is only achieved when the semantic domains are identical

  - hard to guarantee
  - standardisation again....

Theoretical Underpinnings – Ontology Evolution



ologies may be given but...

- Individual tagging
- Folksonomies
- Standardisation
  - Informal and implicit
  - Formal and explicit
- Ontology
  - Ontology emergence
- Semantic Grounding
  - semantic evolution, change and emergence

Theoretical Underpinnings - Logic

Description logics, normality, soundness, completeness, decidability and monotonicity are not sufficient

- Information needs to be removed
- Not all agents "think" in the same way
- Logic varies according to ontology and semantics
- Unknown values not always interpretable as undefined

Logics will vary according to space and even be modified on a per-agent basis

- areas of research:
  - non-monotonicity and defeasilibity
  - multi-valued logics
  - consideration and interpretation values such as $\perp$
  - non-insistence of completeness and decidability
  - etc

Theoretical Underpinnings – Belief, Truth, Consistency

Assertion of information by an agent does not imply truth.

Theoretical Underpinnings - Belief, Truth, Consistency

agent$_1$ asserts

which conforms to:

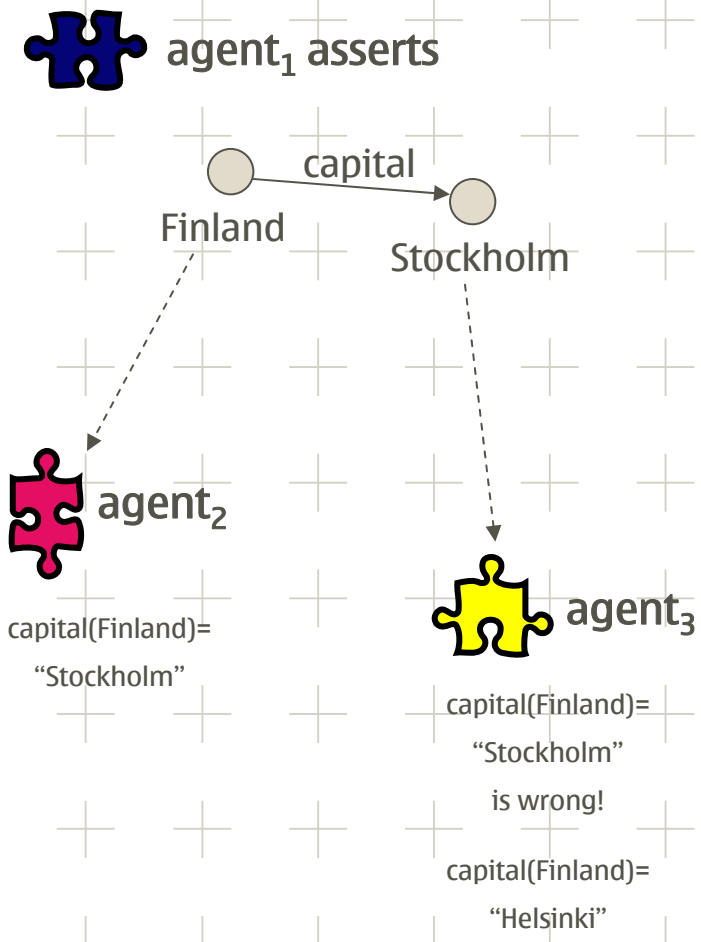
We do not enforce consistency according to ontology

- some spaces might...

Another agent might add additional, contradictory information

- this might be its intent
- interpretation is left to the reader

Theoretical Underpinnings - Belief, Truth, Consistency



Assertion of information by an agent does not imply truth.

Agents 2 and 3 can interpret this according to their beliefs and make decisions accordingly...

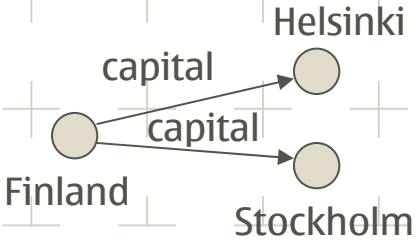
...however mixed they are...

Theoretical Underpinnings - Belief, Truth, Consistency

agent$_1$ asserts:

agent$_2$ asserts:

giving:

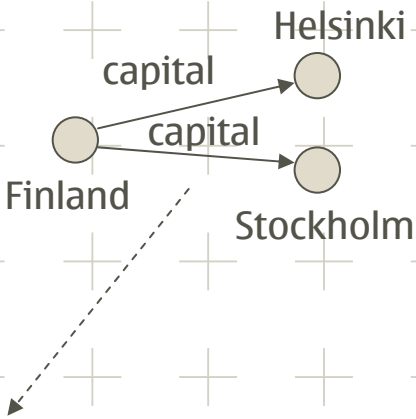


We do not enforce consistency according to ontology

- some spaces might...

which does not conform to:

Theoretical Underpinnings - Belief, Truth, Consistency



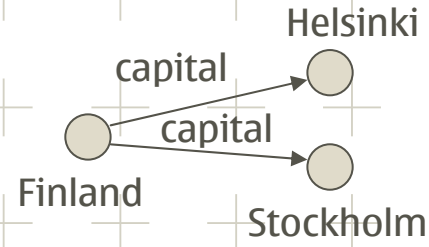
While no decision procedure exists to conclusively choose an answer, there are options.

agent$_3$ interprets locally, with the possible answers:

1. Stockholm is the capital of Finland
2. Helsinki is the capital of Finland
3. Both Stockholm and Helsinki are the capitals of Finland
4. Error
5. Unknown
6. Undefined
7. X is the capital of Finland, where x is not Stockholm nor Helsinki but some answer or interpretation that Agent$_3$ wants to give

- Agents may employ belief revision and truth maintenance algorithms to clean-up such information not adhering to known ontologies
- but this is not always desirable.

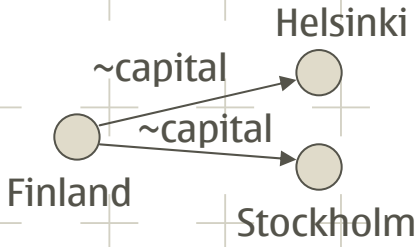
Theoretical Underpinnings - Modality



RDF defines a graph as a distributed conjunction of predicates

capital(Finland, Helsinki)
^
capital(Finland, Stockholm)

Theoretical Underpinnings - Modality



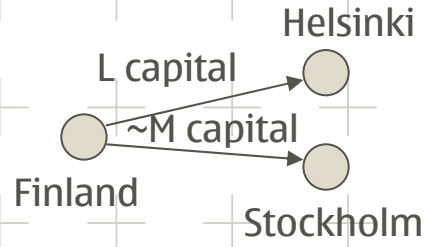
Weakening of this scheme allows inconsistencies

Open or Closed-World ?

~capital(Finland, Helsinki)
^
~ capital(Finland, Stockholm)

What's the capital of Finland?

Theoretical Underpinnings - Modality



Lp = necessarily p

Mp = potentially p

Lp = ~M~p

in some modal systems

Necessity and Potentiality

For some defintion of the above

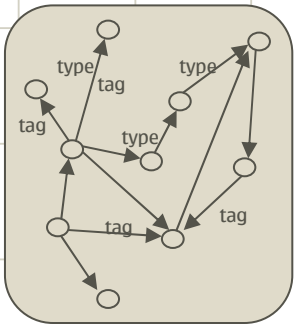
- Linguistic?
  - ·moral obligation
- Mathematical
  - ·S5, S4…other systems

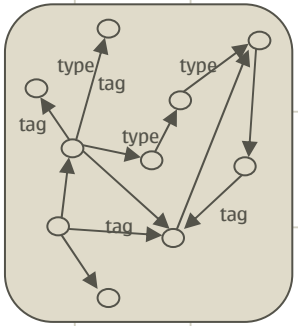
L capital(Finland, Helsinki)

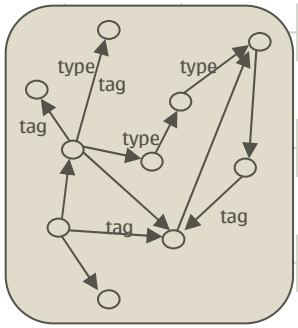
^

~M capital(Finland, Stockholm)

Theoretical Underpinnings - Spaces



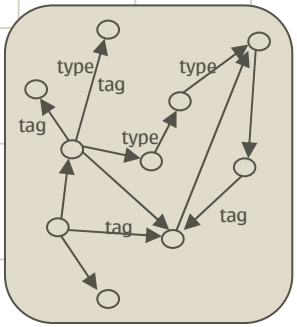


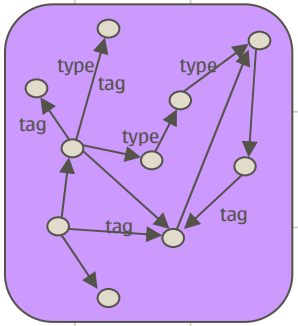


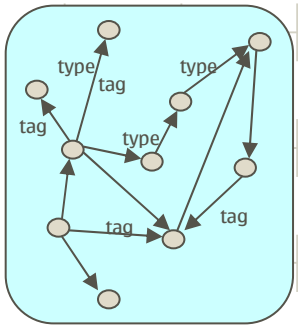
Individual graphs of information are localised as spaces

this is the partitioning of the “Giant Global Graph” concept into more localised and personal spaces.

Theoretical Underpinnings - Spaces



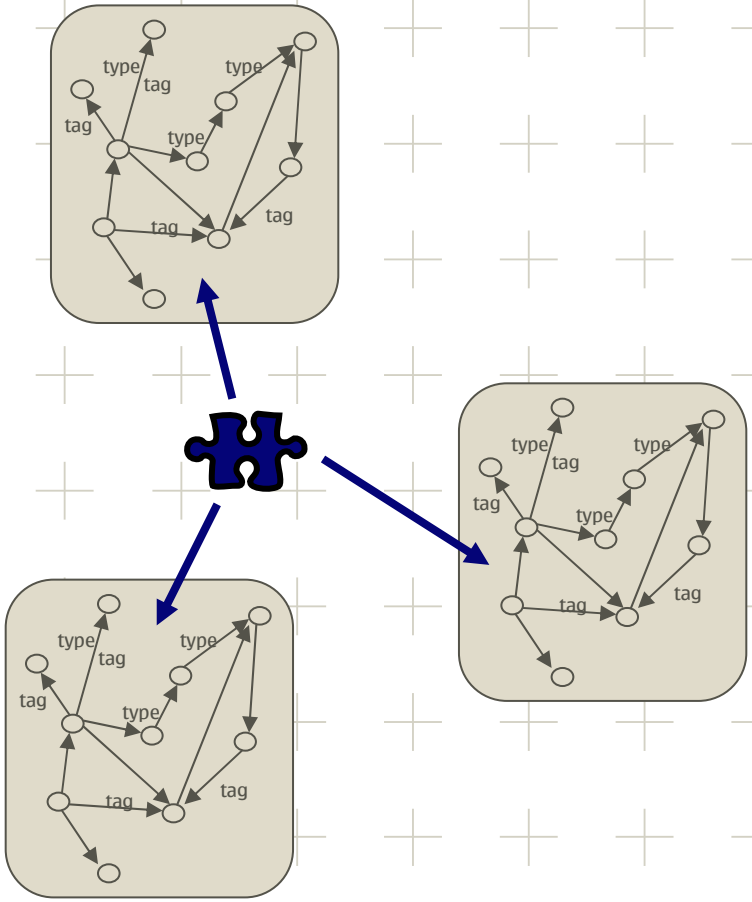




Individual graphs of information are localised as spaces

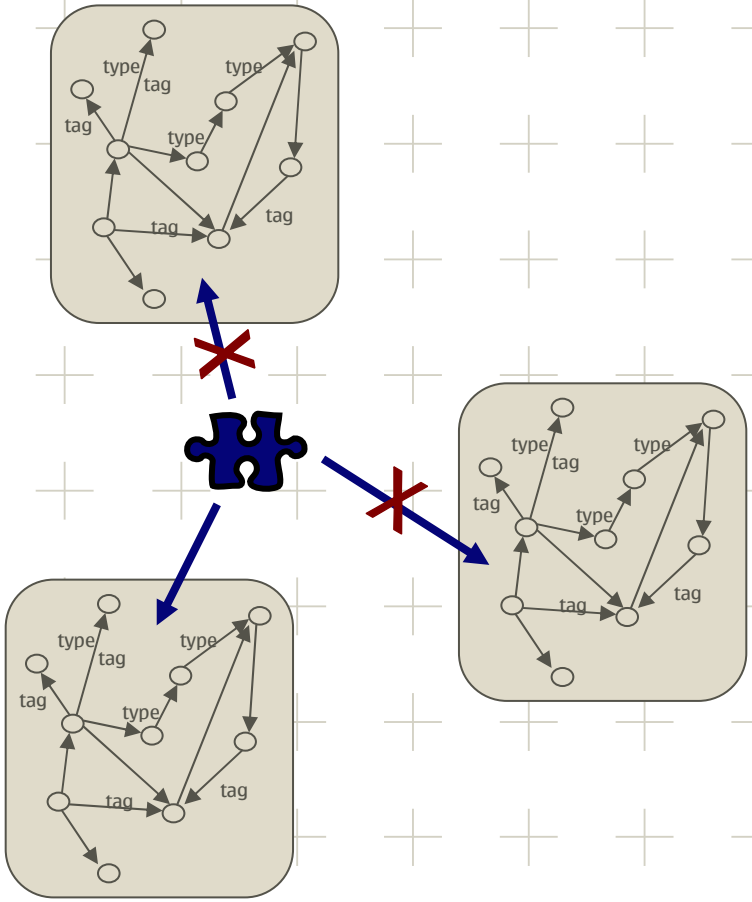
each space may contain its own set of reasoning capabilities and logic for processing the given information

Theoretical Underpinnings – Agents and Spaces



An agent may connect simultaneously to many spaces in order to gather the information it needs to reason over

Theoretical Underpinnings – Agent-Space Membership

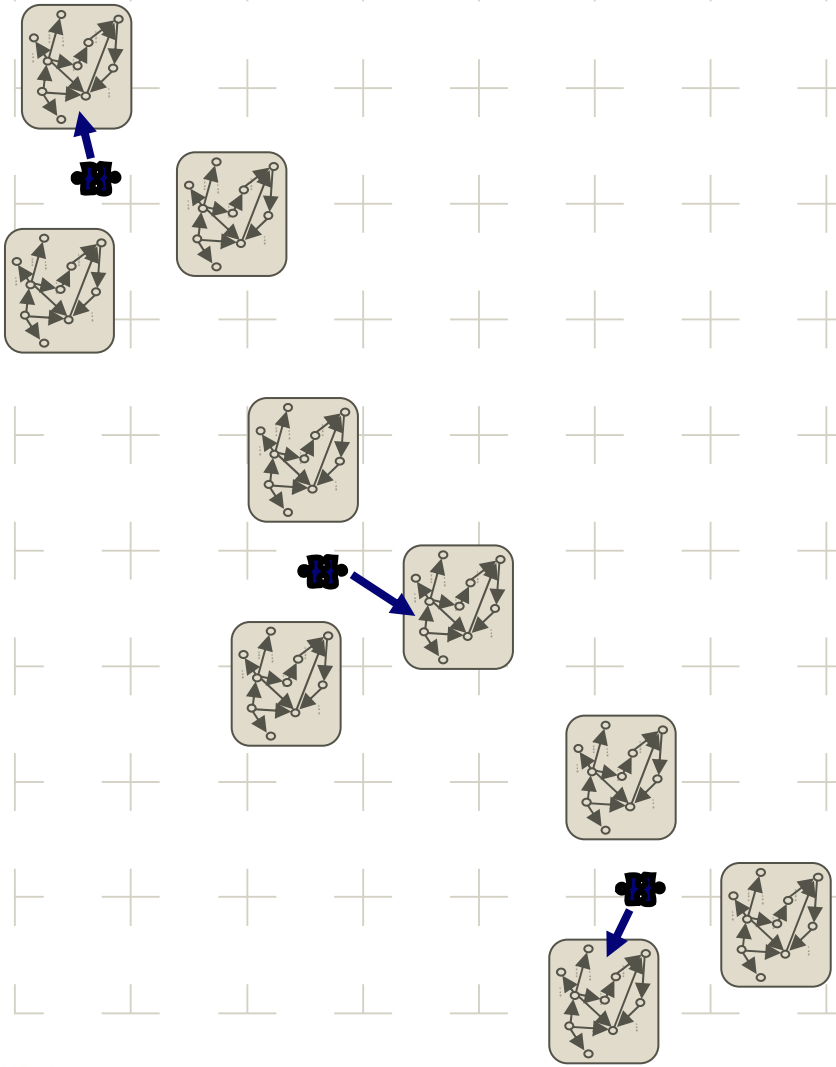


Demarcation of Spaces according to local policy to restrict agent access

Demarcation can be potentially a combination of:

- agent identity
- user identity
- location
- temporal characteristics
- keys (traditional security)
- etc...

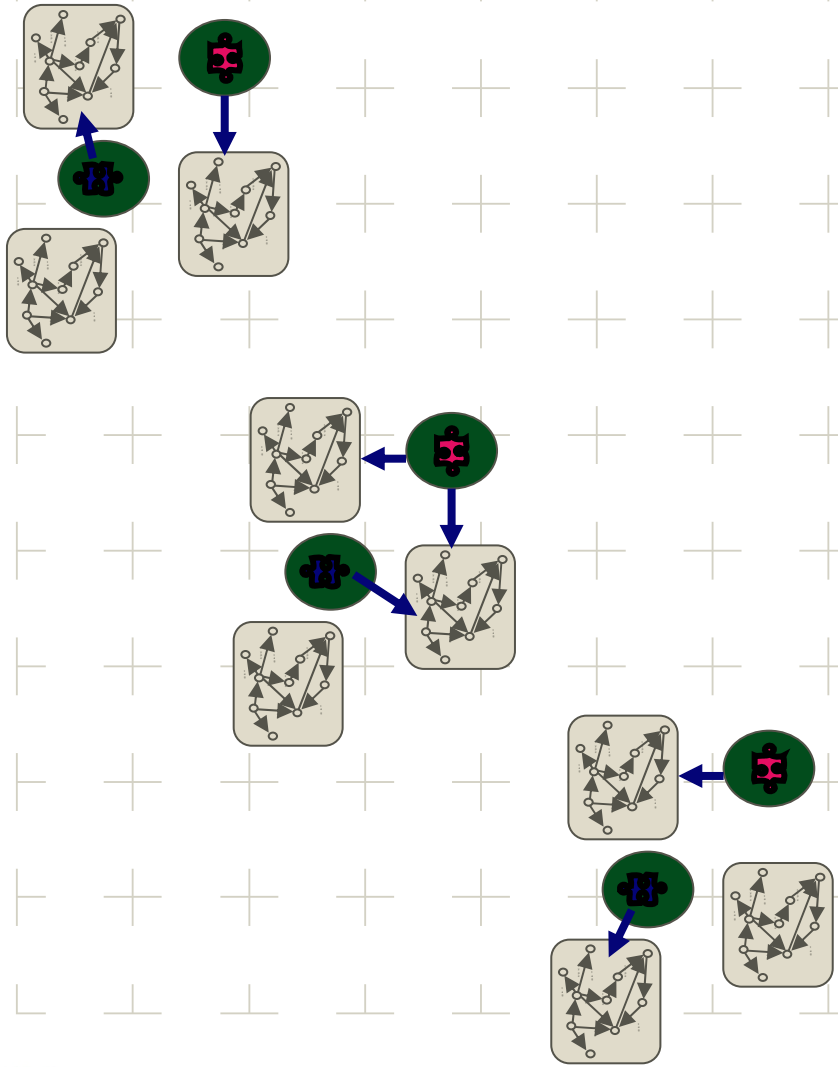
Theoretical Underpinnings – Agent Mobility



Agents are mobile by way of links to spaces

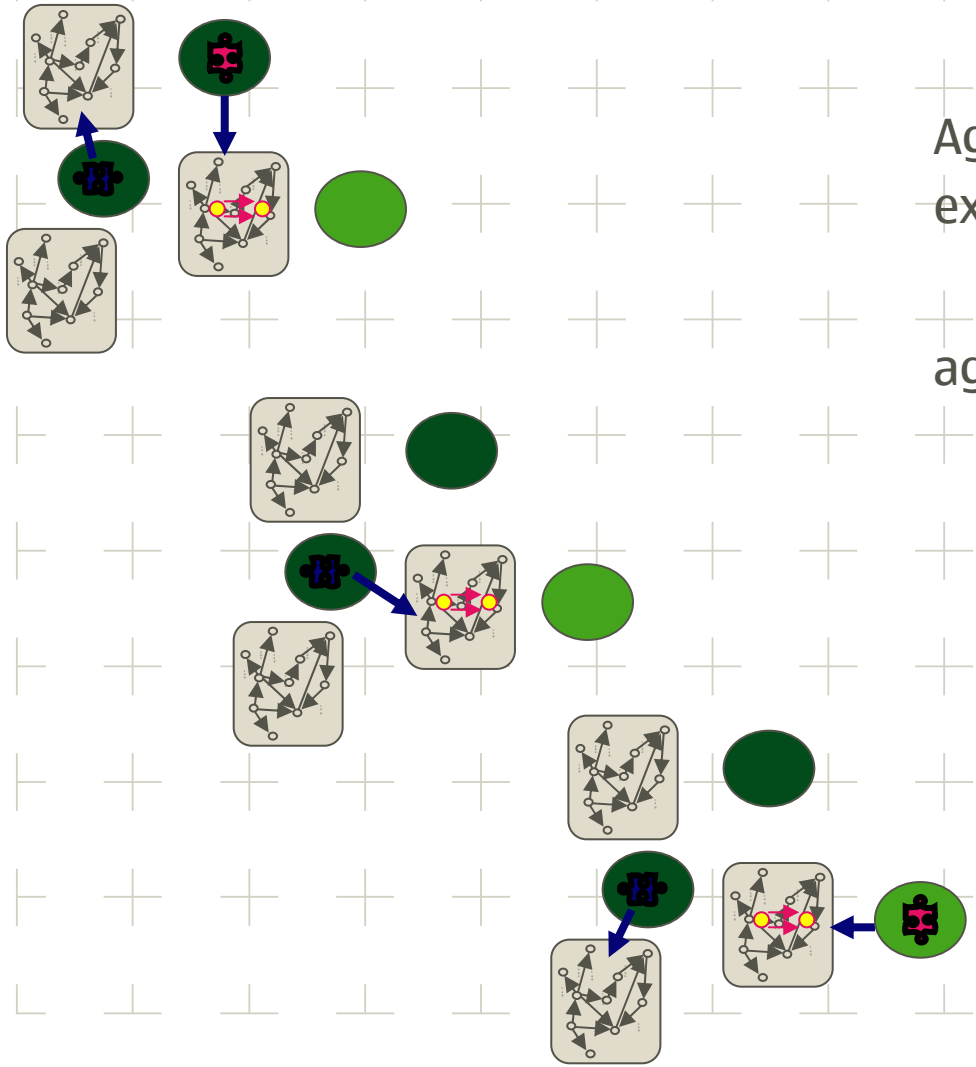
- cf: pi-calculus notions of mobility
- agents are mobile amongst spaces

Theoretical Underpinnings – Agent Mobility



Agents are atomic entities which execute on a single device at a time

Theoretical Underpinnings – Agent Mobility



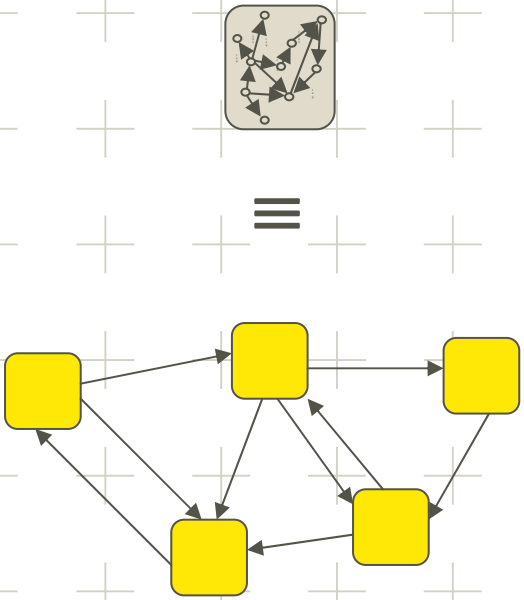
Agents are atomic entities which execute on a single device at a time

agent exist through spaces

- current implementation does not admit mobility of executable code, but...an agent may save its state to a space which another agent might use

- agent existence persistence

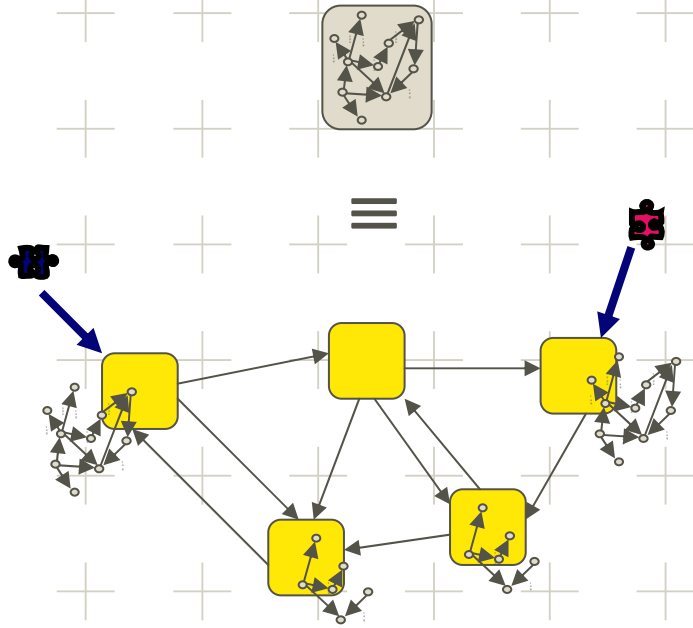
Theoretical Underpinnings – Space Structure



Spaces “just exist”

- represented by one or more Semantic Information Brokers (SIB)
- which are “totally routable”

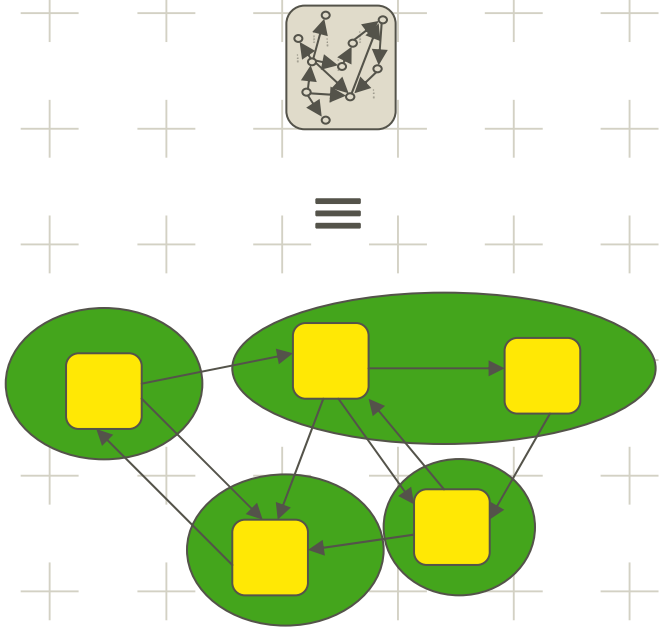
Theoretical Underpinnings – Space Structure



Spaces "just exist"

- each space contains (cf: architecture)
  - ·connectivity functionality
  - ·information storage
    - ·full, partial or even none!
  - ·query distribution and information store synchronisation
  - ·deductive closure calculation mechanisms
- agent always gets a single, consistent view of all information

Theoretical Underpinnings – Spaces and Devices

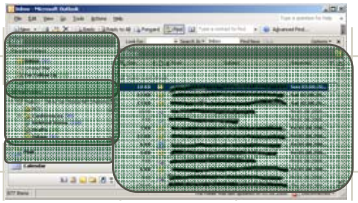


Nominally a SIB executes atomically on a single device

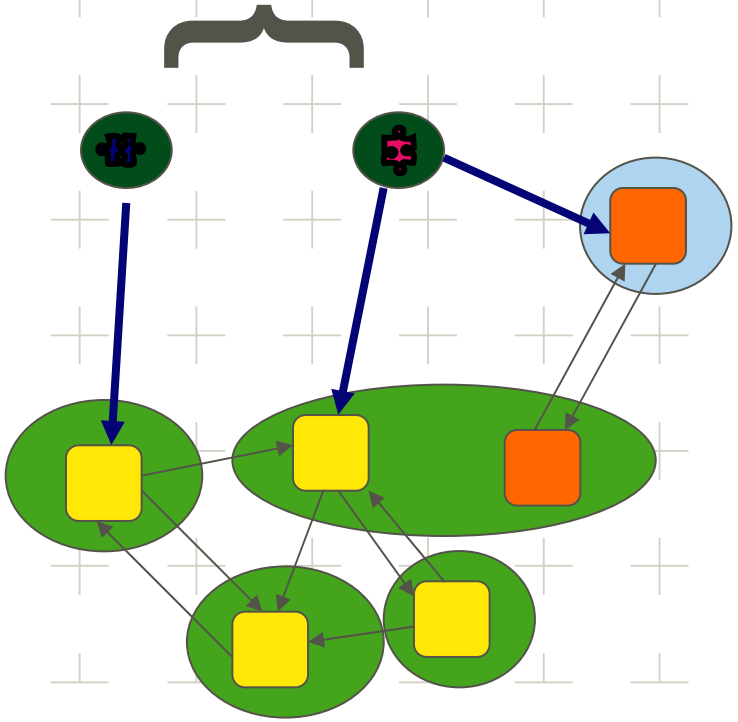
A device may host any number of SIBs

- even ones representing the same space
- SIBs may have different storage and processing capabilities depending upon the hosting device
- the capabilities of a space is given by the union of all the capabilities of the individual SIBs representing that space

Theoretical Underpinnings – Device Abstractions



emerges from



Because applications emerge from agents and spaces emerge from SIBs we abstract the traditional or legacy notion of application completely from its physical presence in any device

- even within the UI the composition of an application is abstracted away from the agents themselves

Implementation and Distribution

- Example Demo Setup

- Python
  - Python 2.5.1 under Linux, Unix, Windows, Symbian etc
- C
  - Linux (N800/N810)
- OpenC
  - Symbian (Nokia N and E series devices)
- Java (4Q08, 1Q09)
  - J2ME, J2EE

Simple XML based protocol – specification and reference implementations will be released as open source distribution 4Q08/1Q09 (estimated)

References

- Oliver, Honkola (2008) **Sedvice: A Triple Space Computing Exploration Environment.** Tripcom Workshop, Galway, April 2008

- Oliver, Honkola (2008) **Personal Semantic Web Through A Space Based Computing Environment**, MSW @ ICSC08, Santa Clara, August 2008 (arxiv.org: 0808.1455)

- Oliver, Honkola, Ziegler (2008) **Dynamic, Localised Space Based Semantic Webs**, WWW/Internet Conference, Freiburg, October 2008

Forthcoming:

- **Space Based Semantic Webs**, Journal of Semantic Computation, Sept'08
- **Semantic Computation**, Journal of Semantic Computation, Dec'08

Current Research

- Security
- Policy
- Trust

- Ontology Construction
  - tagging, folksonomies
  - ontology evolution
  - information recycling
  - semantics

- Synchronisation and Co-ordination of agents

- Connectivity Solutions
  - legacy integration

- Reasoning
  - non-monotonic logics
  - description logics
  - planning, AI ...

- Application/Agent Construction
  - tool environments
  - verification/validation strategies

- Distribution
  - query distribution and optimisation
  - distributed deductive closure calculation

The End

Nokia Research Center

NOKIA

Title font Nokia Large Bold, 28 pt

Sub-headline Nokia Sans Wide Regular, 20 pt

- Body text font Nokia Sans Wide Regular, 20 pt
- Bullet points 100% of the text with same color
- Line spacing in body text 1.20 Lines
- When using animations effects use “Appear” or “Fade”. Avoid wild animations and animated GIF files.

Make sure you have the right Nokia fonts installed. You can download the mandatory font package from Nokia Brand Book:
https://www.nokiamediabank.com
(Nokia Office Package - True Type)

Keep it simple

In general, use as little text as possible. The audience does not want to read a text desert and listen to the presenter at the same time.

The text should only accompany or emphasize the presenters words!
Use suitable pictures to support the message.

Example slide for pictures

Make pictures look like real photography. If you can, add a white frame around the picture. By turning the picture by a few degree, you enhance the impression of a photo placed on your canvas. Add some drop shadow, preferably with a picture editing software (PowerPoint rather makes a grey box than a smooth shadow).

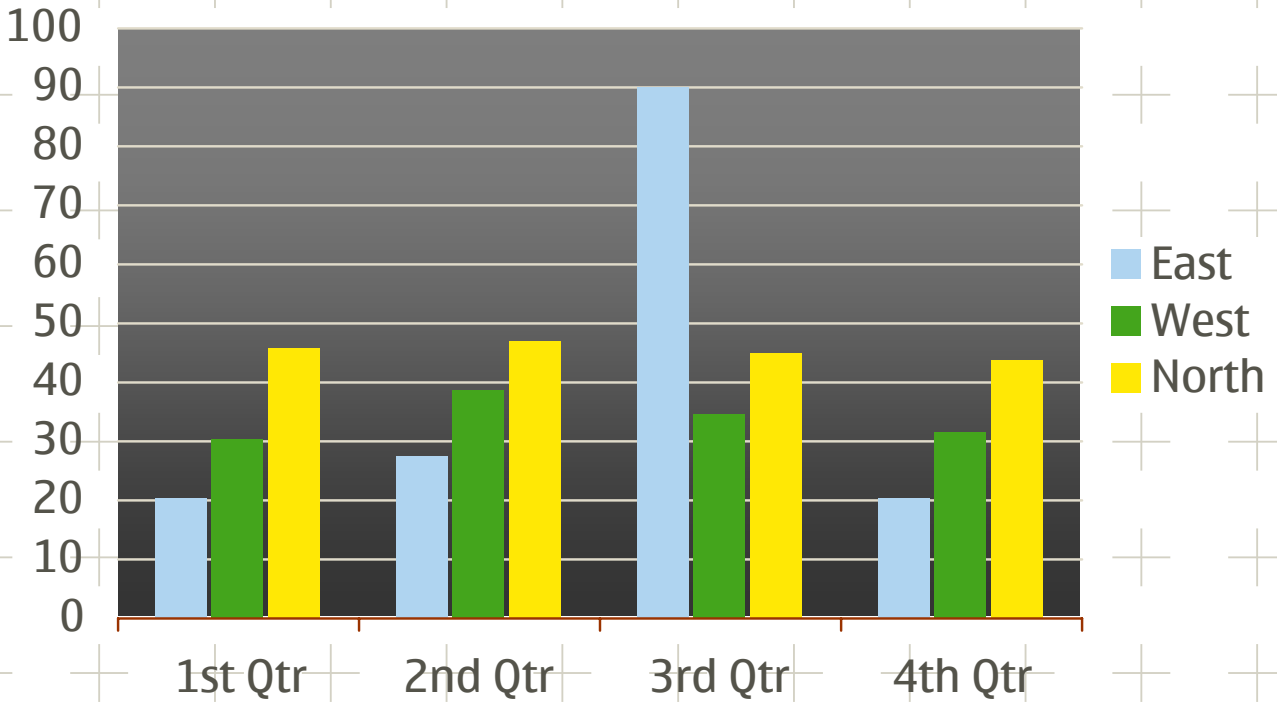
Okay

Good

PowerBoxes

- Consider these color combinations of boxes and text.
- You can use tints of the darker colors.
- Avoid outlines around objects. We don't need them.
- Use boxes with rounded corners using a small radius.
- Use a bit of transparency (20-30%)

Diagrams

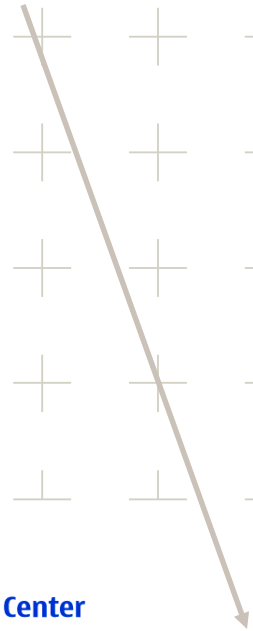


Double click the diagram to edit the numbers.

Editing the Footer

To edit the Footer:

1. Go to Menu > Header and Footer...
2. Replace “First Name Last Name” in the dialog box with yours. Don’t change the Master slide.



Attention when printing:

Make sure that the setting in the printer dialog is set to color - even for black and white printer. Otherwise the background and NRC identifier might not be included in the print.