Service-oriented Distributed Applications in the Future Internet: The Case for Interaction Paradigm Interoperability

Georgios Bouloukakis[1]

Joint work with Nikolaos Georgantas[1], Sandrine Beauche[2], Valérie Issarny[1]

[1] Inria Paris-Rocquencourt, France
[2] Ambientic, France

IFIP WG on Services-oriented Systems, Amsterdam, 11/10/2013

Inria

CHOReOS

SEVENTH FRAMEWORK PROGRAMME

Service-oriented applications in the FI: Search & Rescue (S&R)

life sign sensors

JavaSpaces data space

Middleware interoperability required

planning & deployment

JMEDS Web service

S&R personnel

S&R service choreography

Info diffusion & coordination

JMS event-based system

S&R coordinator

monitoring & control centers

Heterogeneous middleware interaction protocols

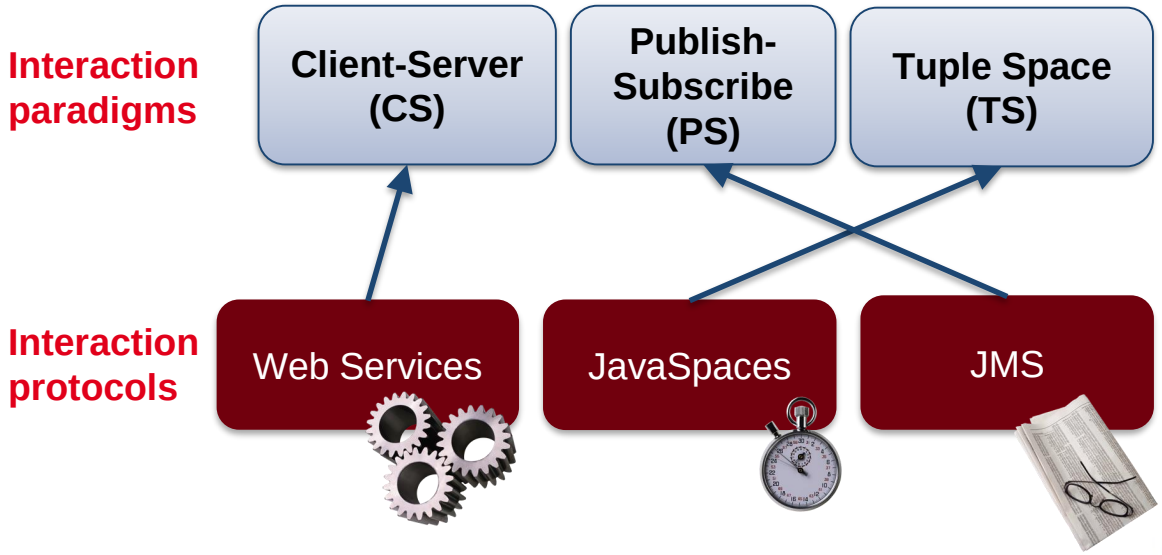
Web Services

JavaSpaces

JMS

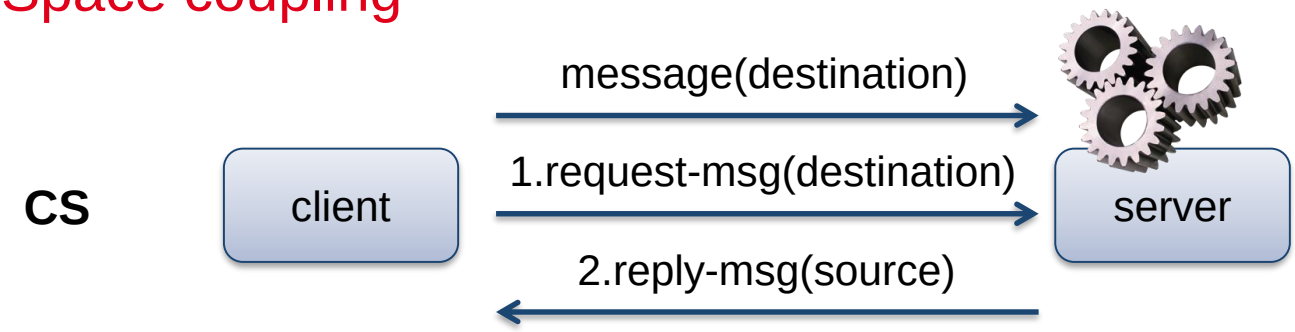
Interaction paradigms in middleware

- Look for comprehensive systematic solution to middleware interoperability
- Classify middleware implementations into *families*
- Families follow well-known *interaction paradigms*

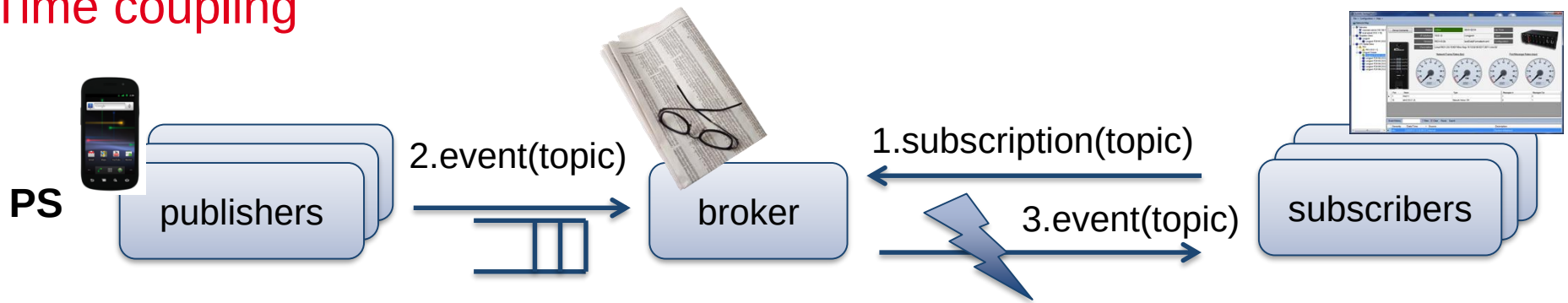


Some informal semantics of interaction paradigms

Space coupling



Time coupling



Concurrency

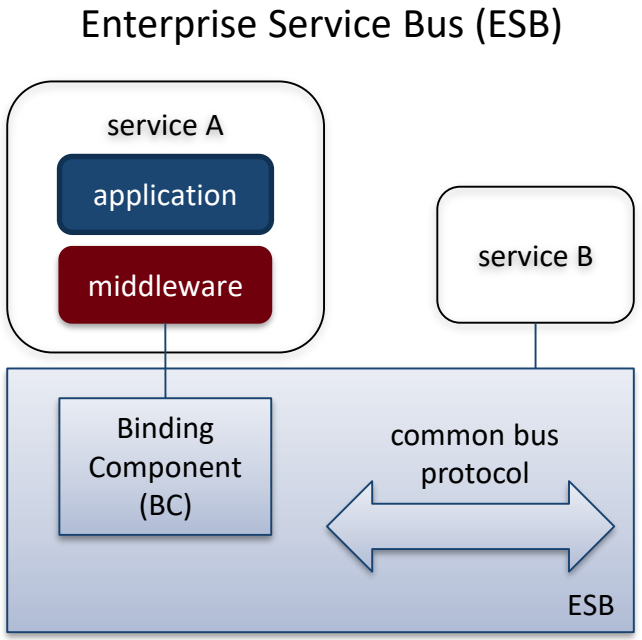
State-of-the-art in interaction paradigm interoperability

Theoretical approaches for individual paradigms

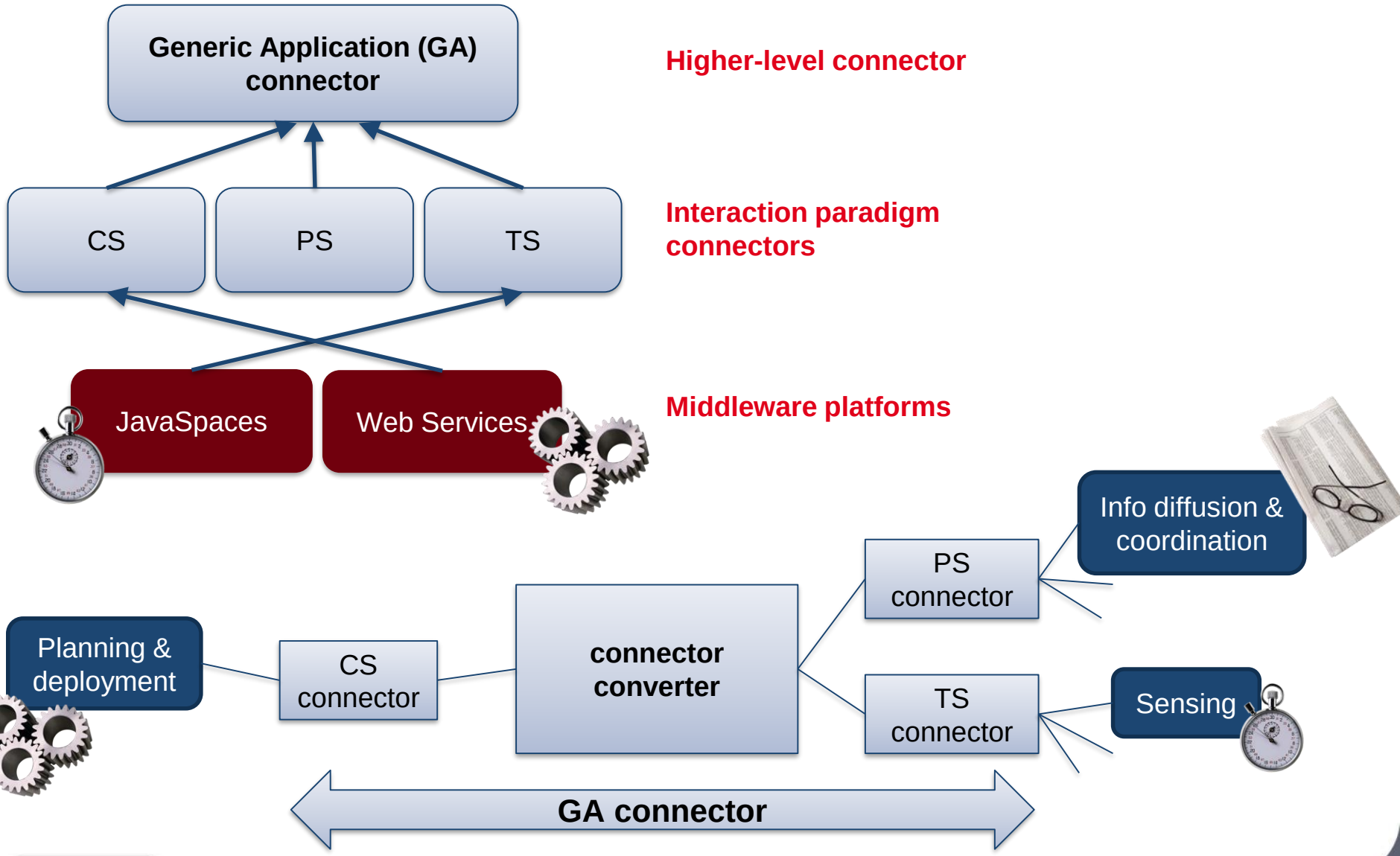
- Rely on concurrency theory, process algebras, architectural connectors
- No study of semantics *across interconnected paradigms*

Practical cross-paradigm approaches

- Typically apply to specific middleware implementations



Solution to middleware interoperability

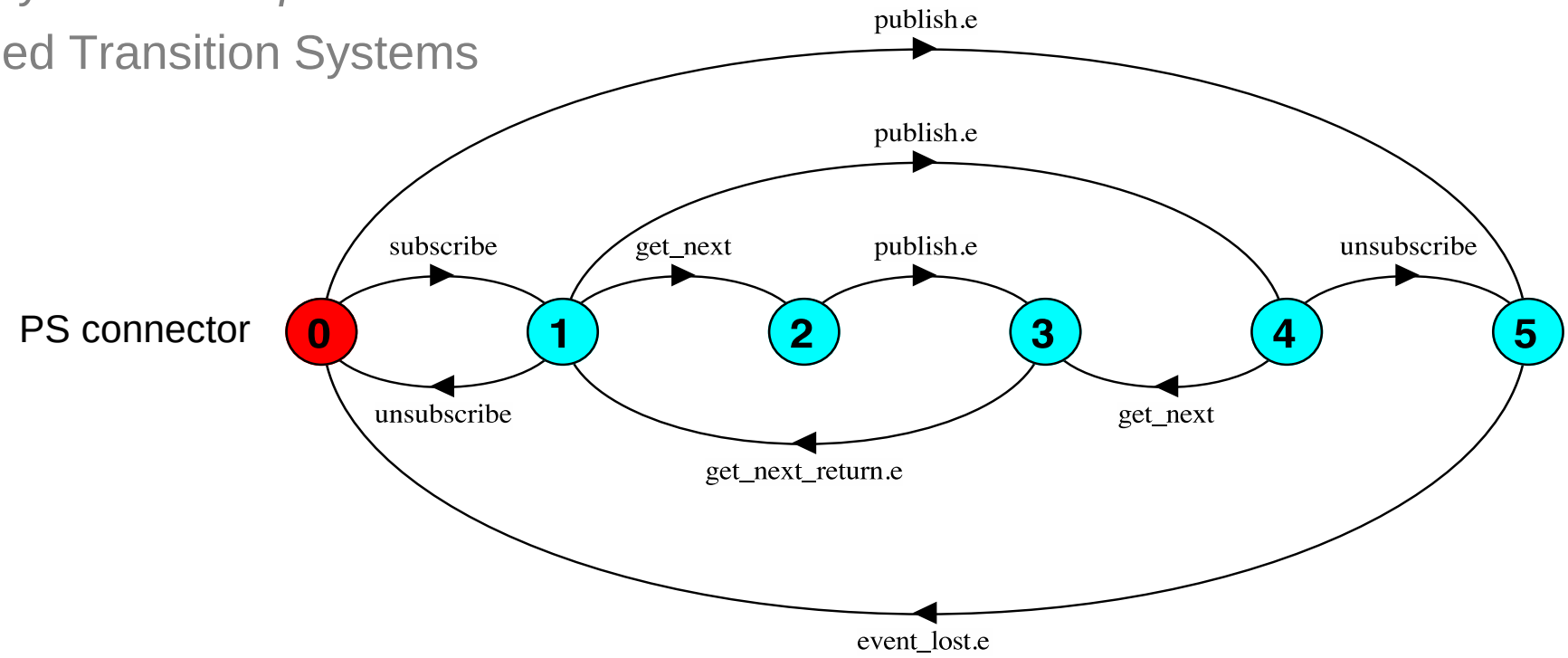


Mapping of space coupling – IDLs

	CS	PS	TS	GA
element	message	event	tuple	data
main scope	system ID	system ID	system ID	system ID
sub-scope	operation	topic	template	*data qualifier*
interaction semantics	{one-way, notification, request-response, solicit-response}	{publish, subscribe}	{write, take, read}	{post, get, post-get, get-post}
S&R element	*in:* {sensorLocation, lifeSign} *out:* {personnelLocation}	*out:* {personnelId, personnelLocation}	*out:* {sensorLocation, lifeSign}	*in/out data*

Time coupling and concurrency of CS, PS, TS connectors

- Specify connector *protocols* with Labeled Transition Systems

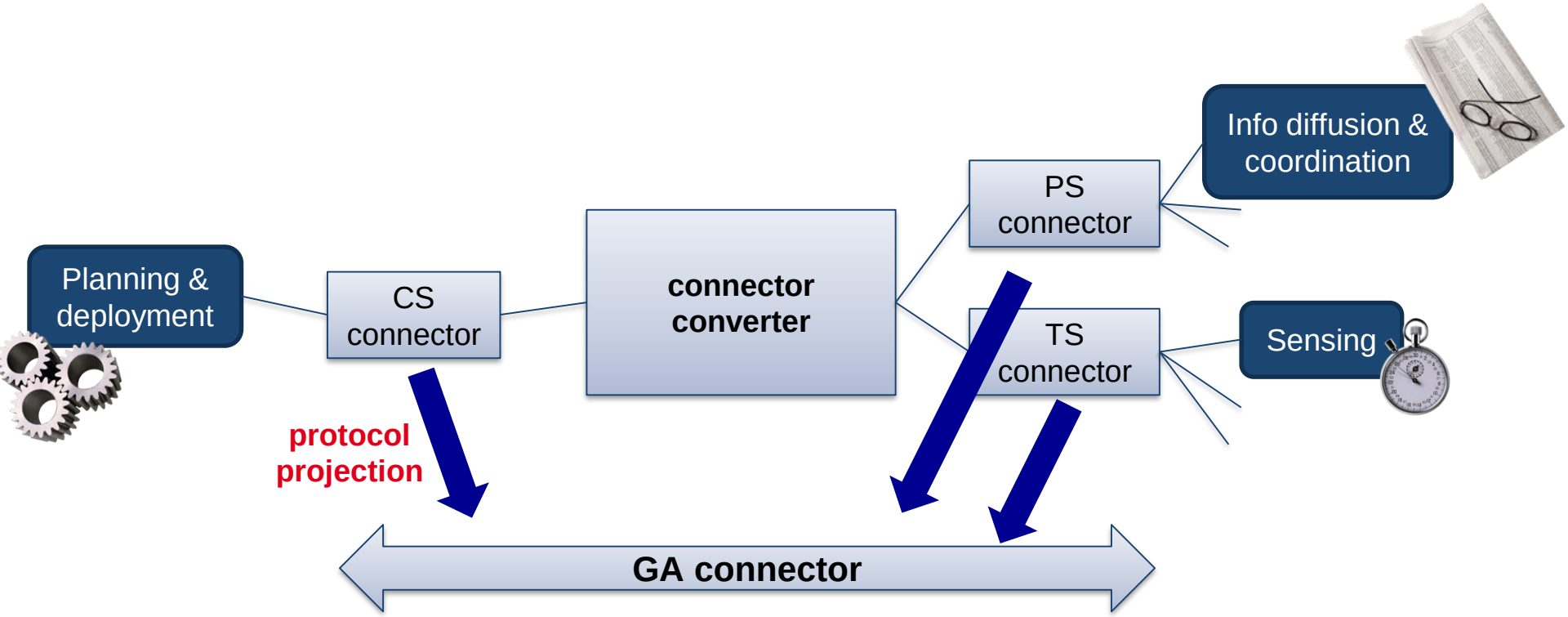


- Express and verify *semantics* in LTL temporal logic

fluent SUBSCRIBED = < {subscribe}, {unsubscribe} >

assert ***EVENT_RECEIVED_IF_SUBSCRIBED_BEFORE_PUBLISH*** *=*
forall [e:EVENT] [] ((SUBSCRIBED && publish[e]) -> (!event_lost[e] U get_next_return[e]))

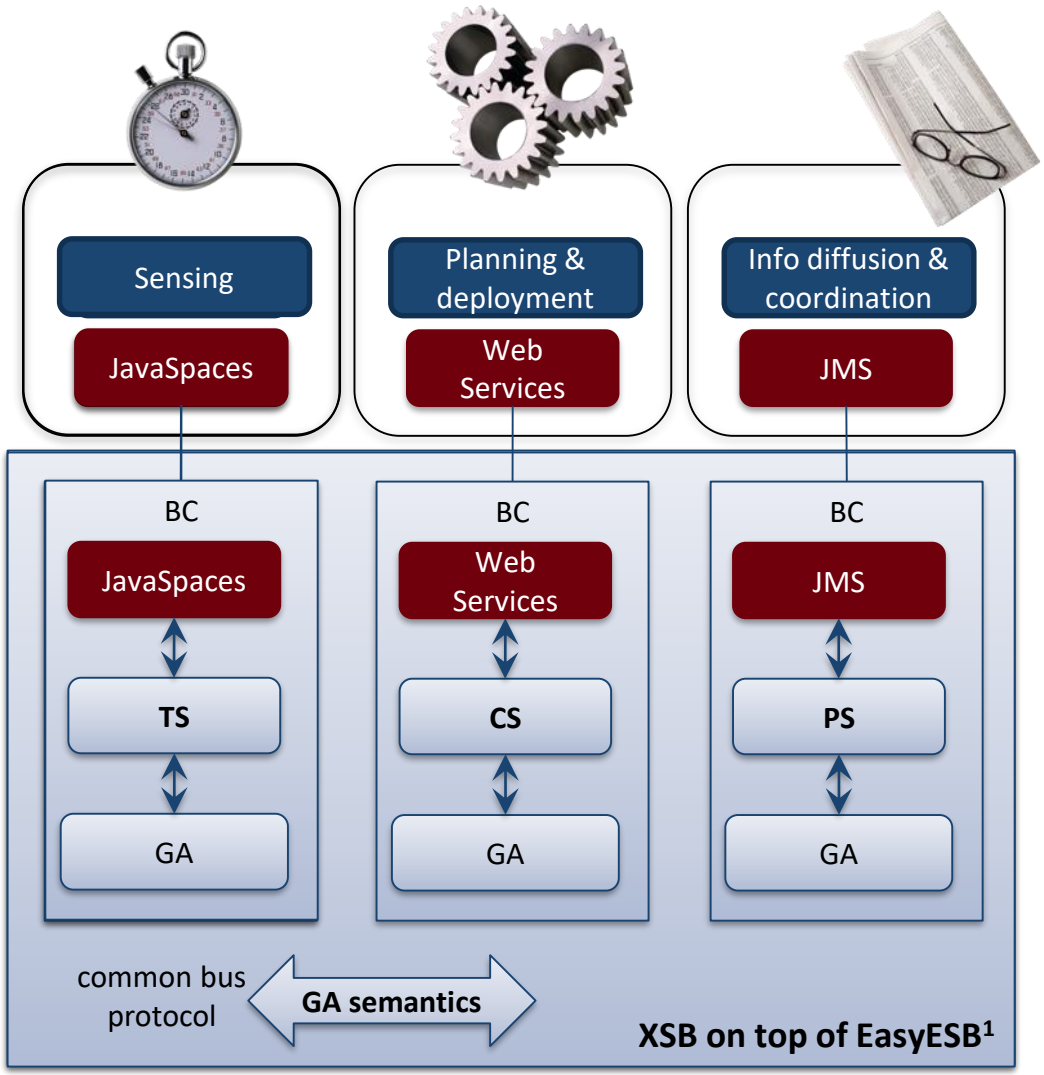
Mapping of time coupling and concurrency



- Rely on the method of *protocol conversion via projections*[1]
- Common semantics of CS, PS, TS apply to GA *end-to-end*

[1] S. S. Lam. Protocol Conversion. IEEE Trans. Softw. Eng. 14, 3. 1988.

Implementation of the GA connector: an eXtensible Service Bus (XSB)

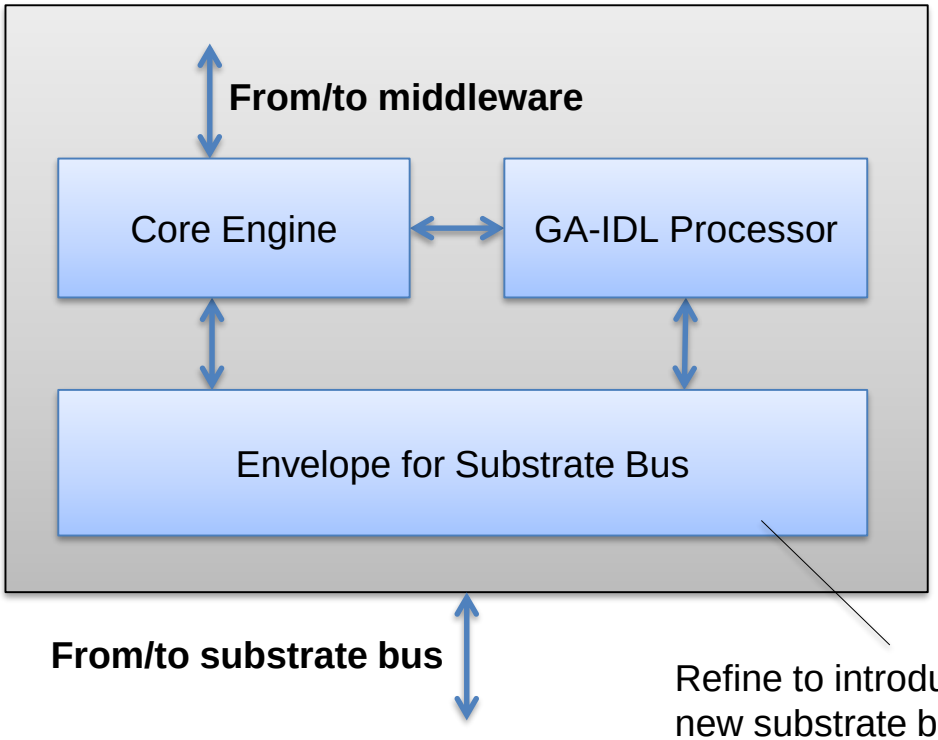


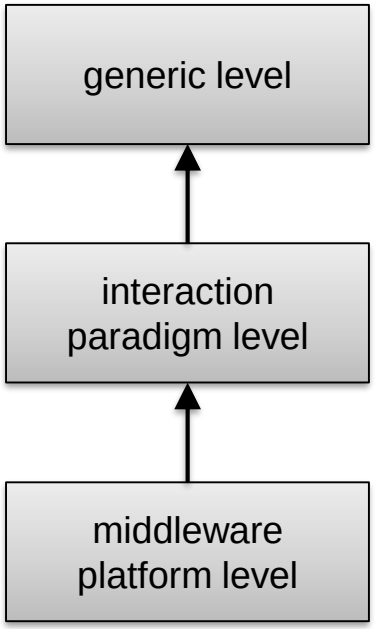
- XSB is a generic bus
- Build on top of a *substrate bus*
- Cross-paradigm integration in the S&R application

[1] https://research.linagora.com/display/easyesb

XSB architectural framework

BC extensible architecture





Refine to introduce a new interaction paradigm

Refine to introduce a new middleware

Refine to introduce a new application service

Evaluation – Effort for S&R application design

Development effort for composing the three available application services into the S&R application

	CS-, PS-, TS-IDL descriptions (XML lines)	Generated GA-IDL descriptions (XML lines)	Data mapping directives (XML lines)
All 3 application services	407	249	226

Evaluation – Extensibility of the XSB framework

Development effort for the JMS binding component

	Lines of code	PS-, GA-IDL schemas (XML lines)	Configuration files (XML lines)
Total effort	18021	2836	248
Developer's effort	1162	191	12
Developer's effort ratio	6%	6%	4%

Evaluation - Performance

- Latency overhead introduced by the bus for all interconnection combinations
- Comparison between XSB and substrate bus

Interconnection	Latency overhead (ms)	Comparison
CS-CS via EasyESB	258	–
CS-CS via XSB	261,5	+1,4%
CS-PS via XSB	283	+9,7%
CS-TS via XSB	276	+7,0%
PS-TS via XSB	298	+15,5%

Conclusion and future perspective

- Service-oriented applications in the Future Internet
  - Require cross-middleware interoperability
- Tackle this challenge via
  - Abstractions and mappings for interaction paradigms
  - Assessment of end-to-end semantics
  - Implementation into an extensible service bus

➢ Next step
  - Extend with support for continuous interactions – data streaming protocols

Thank you

Further information:

XSB: xsb.inria.fr

CHOReOS: www.choreos.eu

Inria ARLES: www.rocq.inria.fr/arles