Web Services Business Process Execution Language (WS-BPEL)

Dieter König, IBM
dieterkoenig@de.ibm.com

Second Annual OASIS Adoption Forum
London, October 17, 2005

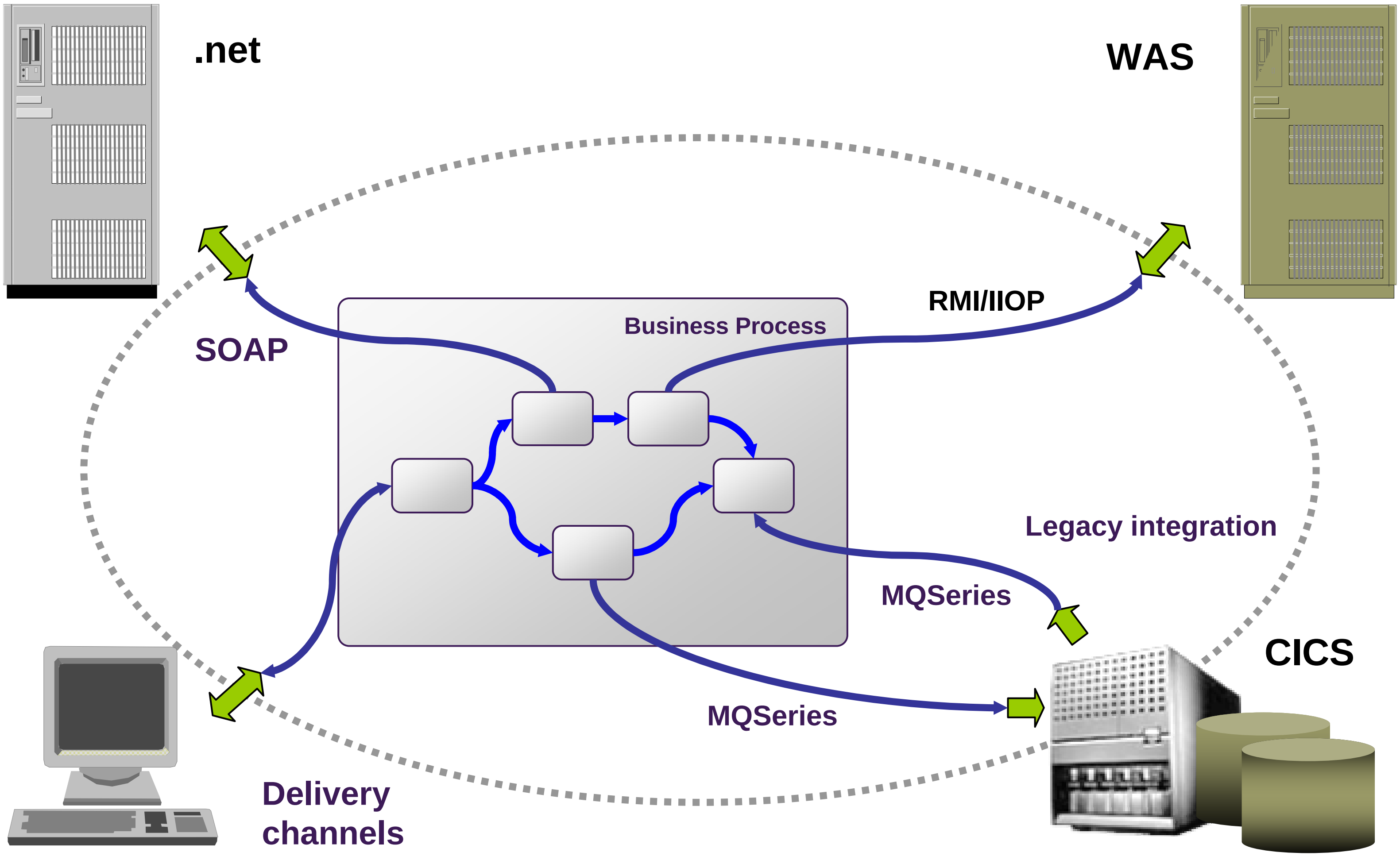
Outline

- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support

Motivation

- Application integration is a key problem facing businesses
  - Intra enterprise integration (Enterprise Application Integration)
  - Integrating with partners (Business Process Integration)
- Web services → move towards service-oriented computing
  - Applications are viewed as “services”
  - Loosely coupled, dynamic interactions
  - Heterogeneous platforms
  - No single party has complete control
- Service composition
  - How do you compose services in this domain?

Application Integration



Two-level Programming Model

- Programming in the large
  - Non-programmers implementing flows
    - Flow logic deals with combining functions in order to solve a more complex problem (such as processing an order)
- Programming in the small
  - Programmers implementing functions
    - Function logic deals with a discrete fine-grained task (such as retrieving an order document or updating a customer record)

Process Usage Patterns

- Aiming for a single approach for both …
    - Executable processes
        - Contain the partner’s business logic behind an external protocol
    - Abstract processes
        - Define the publicly visible behavior of some or all of the services an executable process offers
        - Define a process template embodying domain-specific best practices

Process Model Requirements

- Portability and Interoperability
- Flexible Integration
  - Rich, and easily adaptable to changes in the services it is interacting with
- Recursive, type-based composition, enables …
  - third-party composition of existing services
  - providing different views on a composition to different parties
  - inter-workflow interaction
  - increased scalability and reuse
- Separation and composability of concerns
  - Decoupled from the supporting mechanisms (quality of service, messaging frameworks)
- Stateful conversations and lifecycle management
  - Can carry multiple stateful long-running conversations
- Recoverability
  - Business processes, and in particular long running ones, need a way to build-in fault handling and compensation mechanisms to handle and recover from errors

WS-BPEL

- WS-BPEL enables …
  - Defining business processes as coordinated sets of Web service interactions, recursively into new aggregated Web services
  - Defining both abstract and executable processes
    - Abstract processes for e-commerce specifications
    - Executable processes provide a model to integrating enterprise applications
  - Creating compositions of Web services
    - Composition based on abstract descriptions

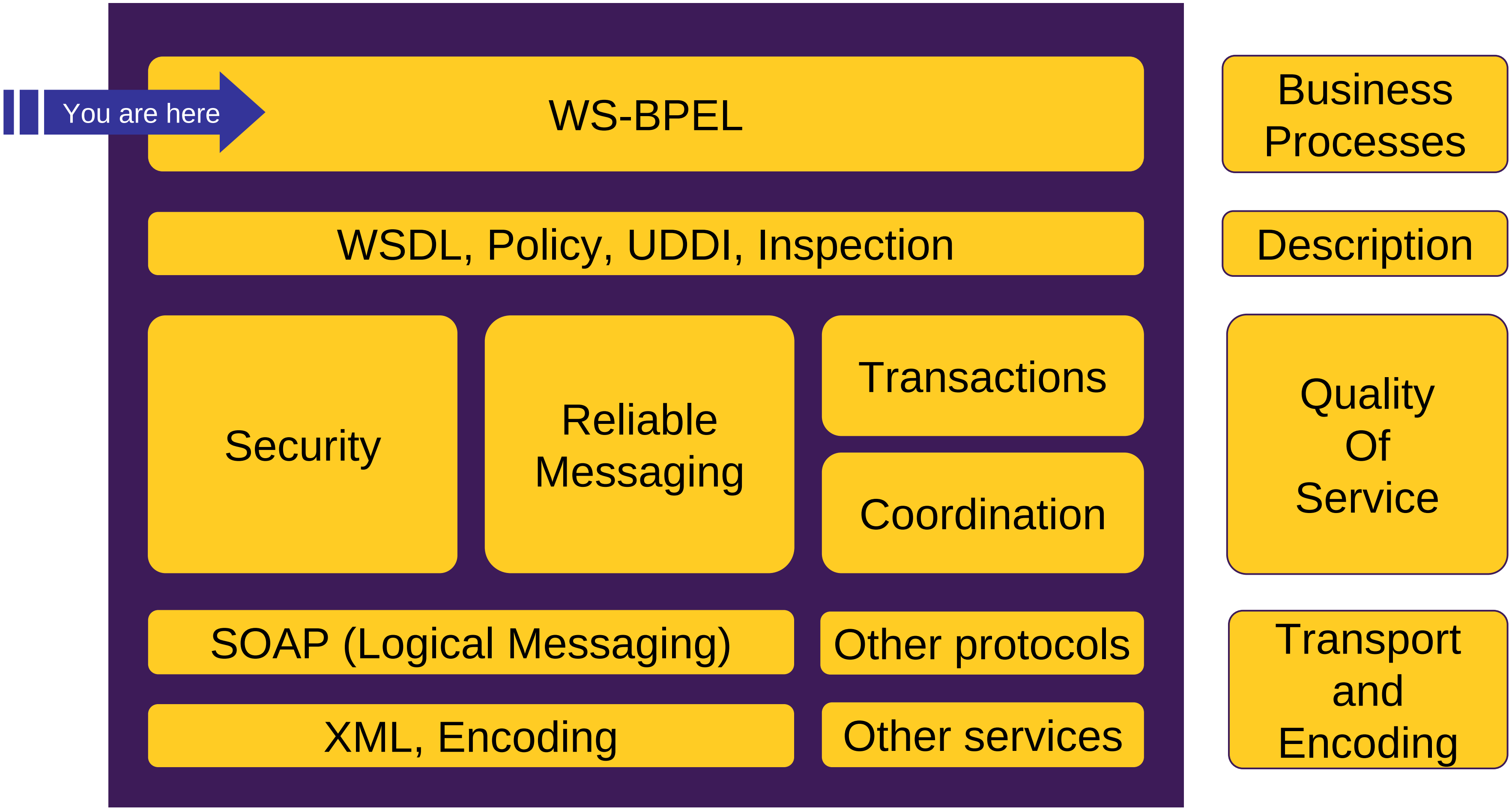
- WS-BPEL provides portable, interoperable process models

- WS-BPEL comes from …
  - Strong roots in traditional flow models
  - Plus many concepts from structured programming languages
  - All laid on top of WSDL and core XML specifications
  - Merges WSFL and XLANG concepts

WS-BPEL Specifications

- BPEL4WS 1.0 (7/2002)
  - Original proposal from BEA, IBM, Microsoft
  - Combined ideas from IBM’s WSFL and Microsoft’s XLANG
- BPEL4WS 1.1 (5/2003)
  - Revised proposal submitted to OASIS
  - With additional contributions from SAP and Siebel
- WS-BPEL 2.0 Committee Draft Specifications
  - Currently in OASIS undergoing standardization

WS-BPEL in the WS-* Stack



You are here

WS-BPEL

WSDL, Policy, UDDI, Inspection

Security

Reliable Messaging

Transactions

Coordination

SOAP (Logical Messaging)

Other protocols

XML, Encoding

Other services

Business Processes

Description

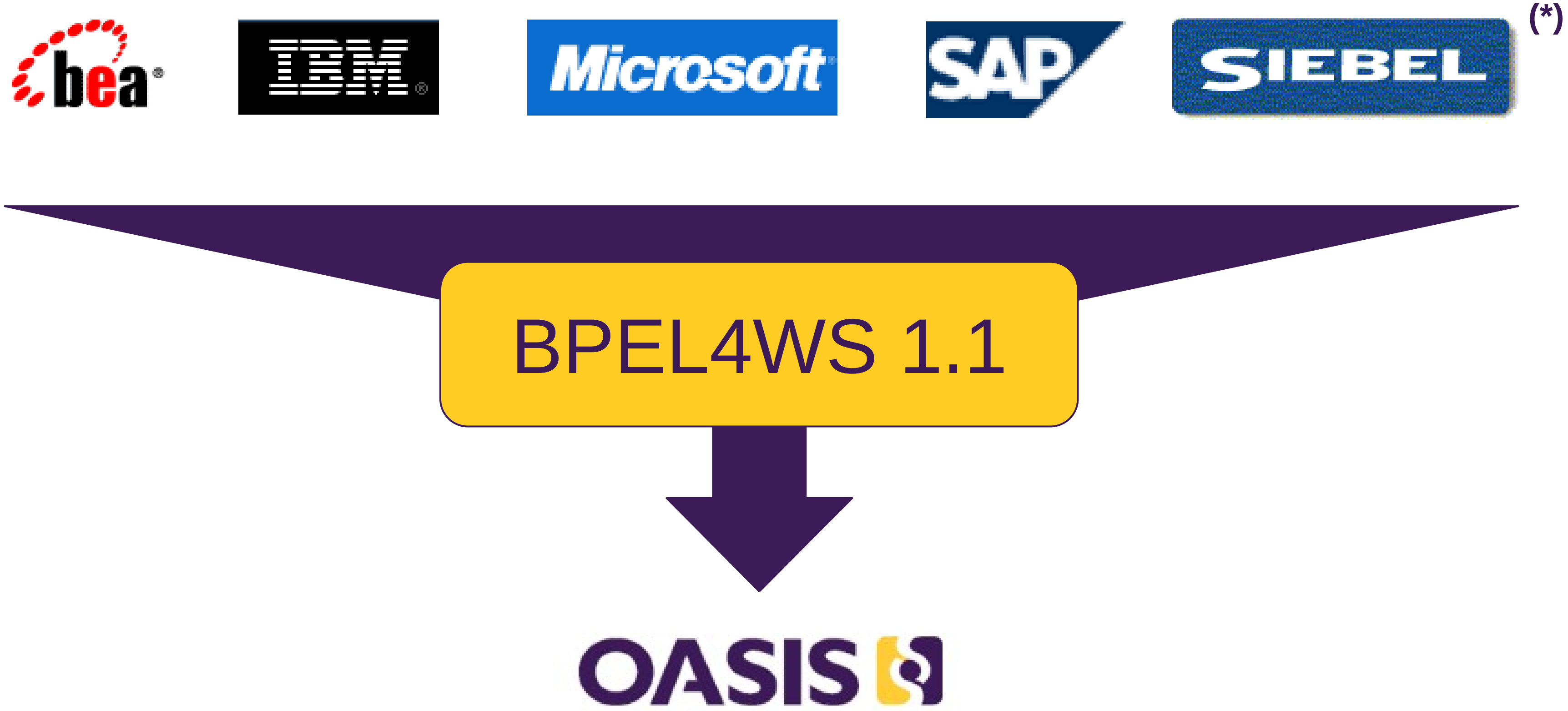
Quality Of Service

Transport and Encoding

Outline

- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support

Getting the Players Together



BPEL4WS 1.1

(*) BPEL4WS 1.1 authors

OASIS Technical Committee

- Over 250 committee members, incl. observers
  - 44 Active voting members, attending weekly calls
- Work on WS-BPEL (TC Charter)
  - Standardize it 🙂
  - Focus on
    - Common concepts for a business process execution language for usage patterns including both the process interface descriptions and executable process models
  - Explicitly **do not** address
    - Bindings to specific hardware/software platforms and other mechanisms required for a complete runtime environment for process implementation

OASIS Technical Committee

- Issues Process
  - List of all issues available at
    http://www.choreology.com/external/WS_BPEL_issues_list.html
  - Issue discussion
    - Weekly calls
    - Quarterly face to face meetings
- Status
  - Deadlines (need 2/3 majority to override)
    - No new feature issues since Aug 15, 2004
    - No new feature issue resolution proposals since April 1, 2005
    - Feature issues that are not resolved are marked as revisitable
  - Latest approved committee draft: September 2005

WS-BPEL Design Goals

- Business processes defined using an **XML-based language**
- **Web services** are the model for process decomposition and assembly
- **The same orchestration concepts** are used for both the **external** (abstract) and **internal** (executable) views of a business process
- Both **hierarchical and graph-like** control regimes are used, reducing the fragmentation of the process modeling space
- An **identification mechanism for process instances** is provided at the application message level
- The **basic lifecycle mechanism** is in implicit creation and termination of process instances.
- A long-running transaction model is defined to support **failure recovery** for parts of long-running business processes
- Language built on **compatible Web services standards in a composable and modular manner**

Outline

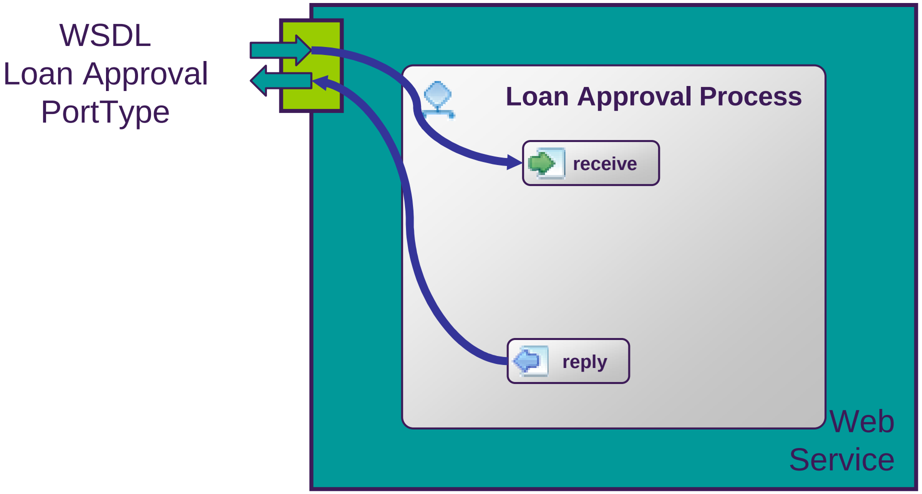
- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support

WS-BPEL Language Structure

- Process
- Partner links
- Data handling
- Properties and correlation
- Basic and structured activities
- Scopes

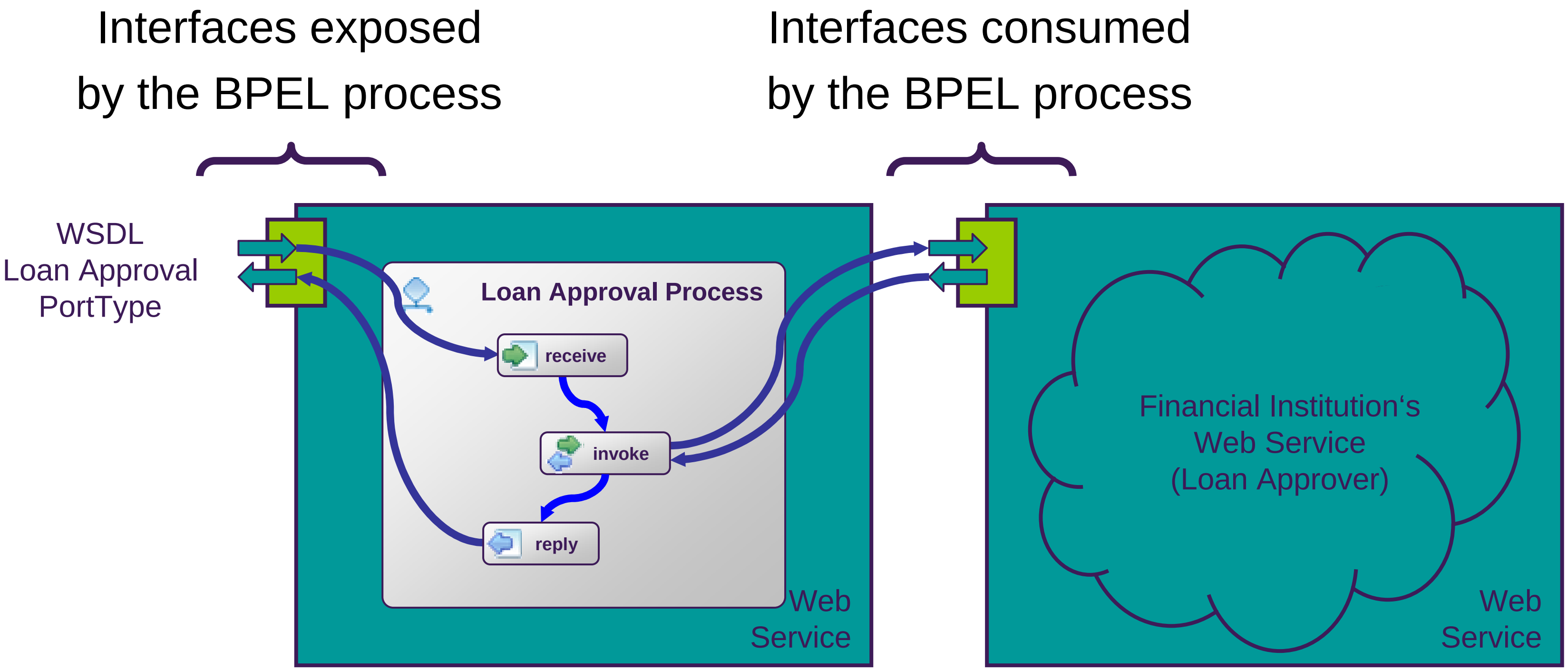
BPEL and WSDL

- BPEL processes are exposed as WSDL services
  - Message exchanges map to WSDL operations
  - WSDL can be derived from partner definitions and the role played by the process in interactions with partners

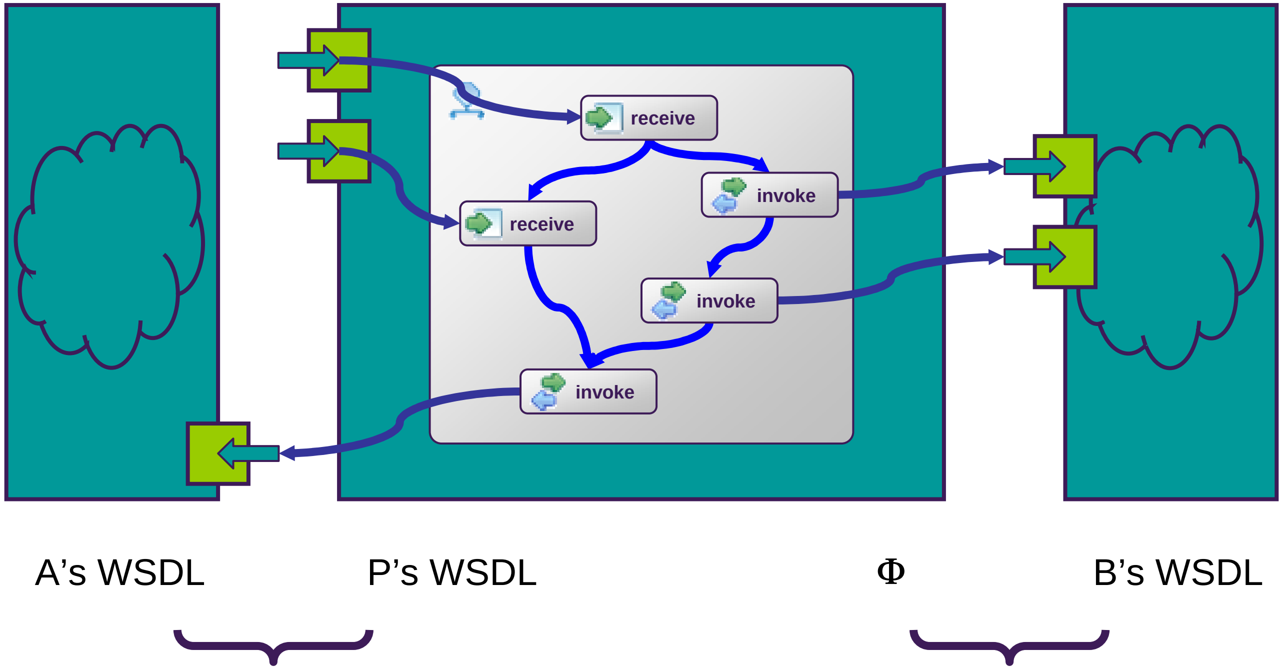


Recursive Composition

- BPEL processes interact with WSDL services exposed by business partners



Composition of Web Services



Service A

Service P

Service B

A’s WSDL

P’s WSDL

Φ

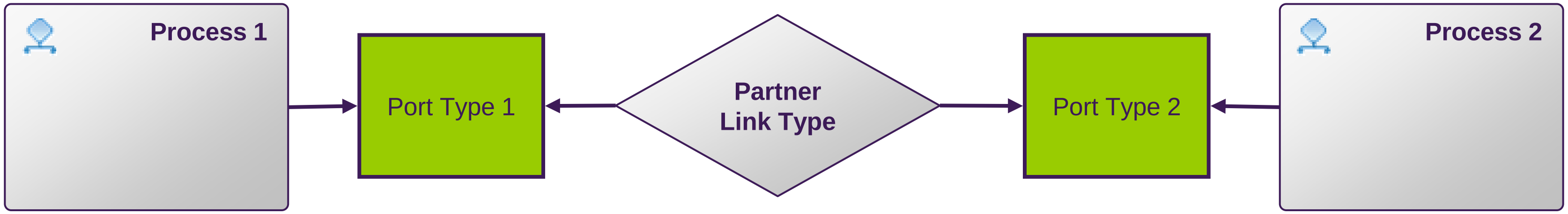
B’s WSDL

Partner Link Type

Partner Link Type

Partner Links

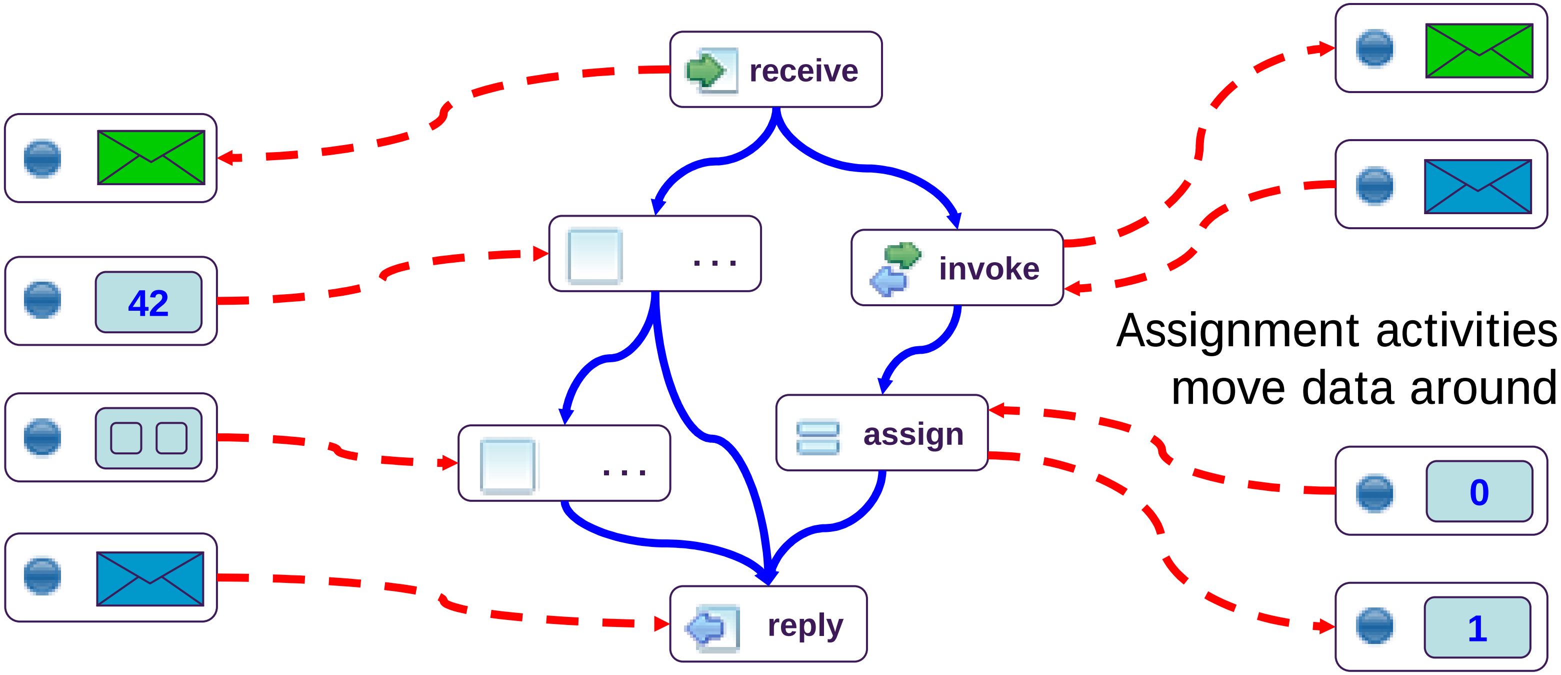
- Partner link: instance of typed connector
  - Partner link type specifies required and/or provided portTypes
  - Channel along which a peer-to-peer conversation with a partner takes place



BPEL Data Model: Variables

Scoped variables typed as
WSDL messages or
XML Schema elements/types

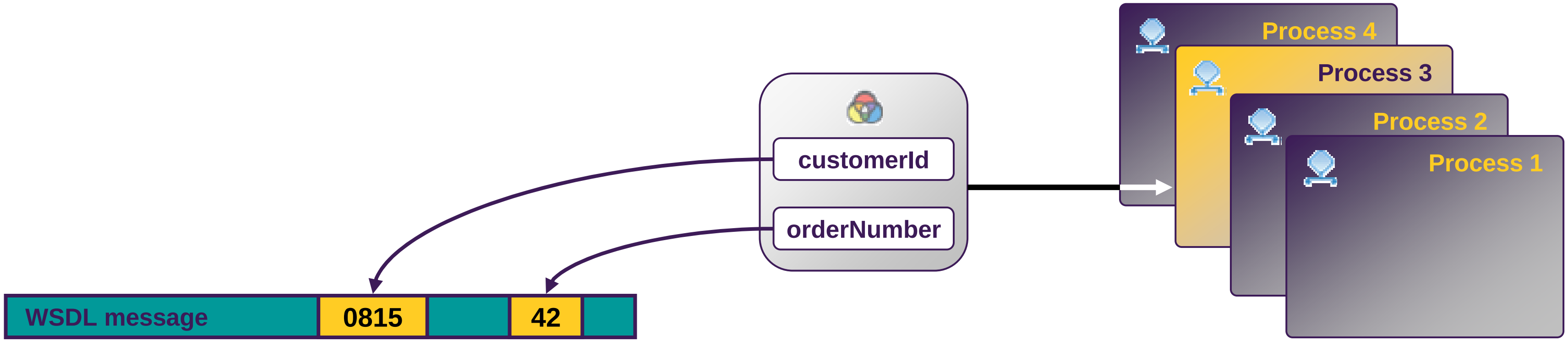
Activities' input and output
kept in scoped variables



Assignment activities
move data around

Properties and Correlation

- Messages in long-running conversations are correlated to the correct process instance
  - Typed properties defined in WSDL are named and mapped (aliased) to parts of several WSDL messages used by the process



Basic Activities

receive

Do a blocking wait for a matching message to arrive

reply

Send a message in reply to a formerly received message

invoke

Invoke a one-way or request-response operation

assign

Update the values of variables or partner links with new data

validate

Validate XML data stored in variables

empty

A “no-op” instruction for a business process

throw

Generate a fault from inside the business process

rethrow

Forward a fault from inside a fault handler

exit

Immediately terminate execution of a business process instance

wait

Wait for a given time period or until a certain time has passed

compensate

Invoke compensation on an inner scope that has already completed

Structured Activities

flow

Contained activities are executed in parallel, partially ordered through control links

if then else

Select exactly one branch of activity from a set of choices

while

Contained activity is repeated while a predicate holds

repeatUntil

Contained activity is repeated until a predicate holds

pick

Block and wait for a suitable message to arrive (or time out)

forEach

Contained activity is performed sequentially or in parallel, controlled by a specified counter variable

sequence

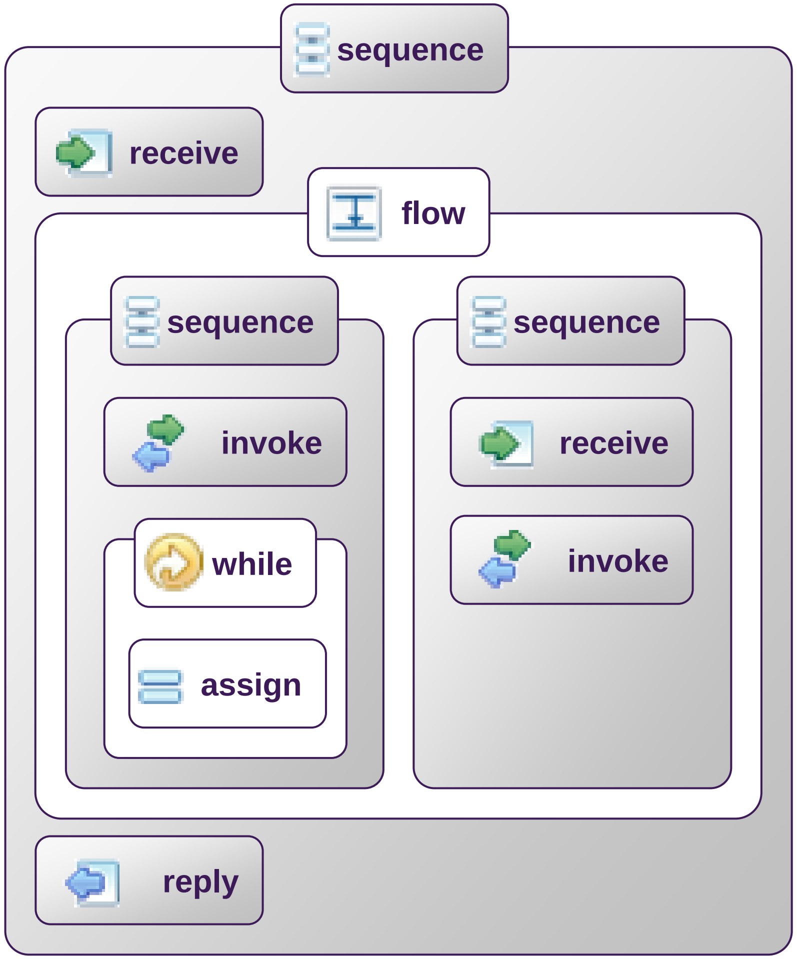
Contained activities are performed sequentially in lexical order

scope

Associate contained activity with its own local variables, fault handlers, compensation handler, and event handlers

Nesting Structured Activities

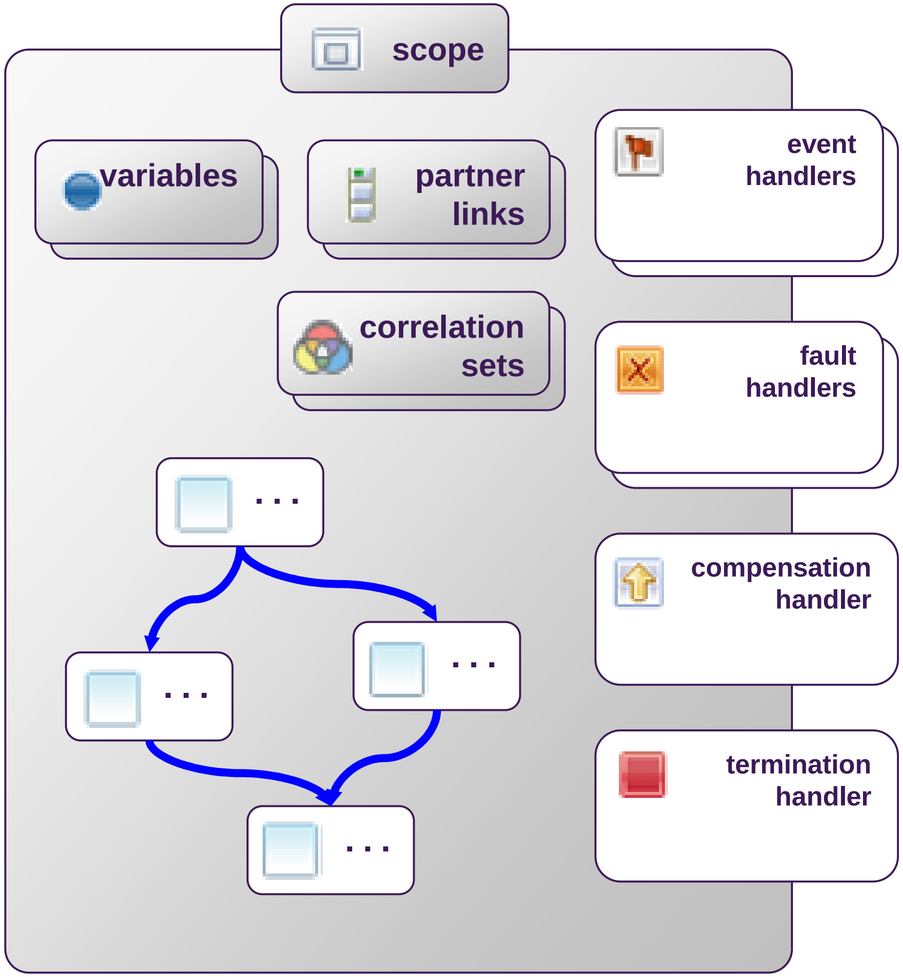
```
<sequence>
   <receive .../>
   <flow>
      <sequence>
         <invoke .../>
         <while ... >
            <assign>...</assign>
         </while>
      </sequence>
      <sequence>
         <receive .../>
         <invoke ... >
      </sequence>
   </flow>
   <reply>
</sequence>
```



Scopes and Handlers

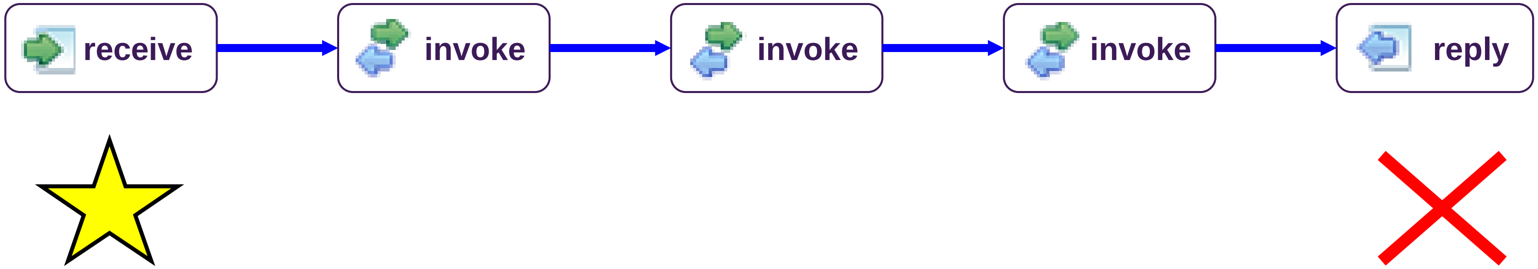
- Scope
  - Set of activities (basic or structured)
  - Local variables
  - Local correlation sets
  - Local partner links

- Handlers
  - Event handlers
    - Message events or timer events (deadline or duration)
  - Fault handlers
    - Dealing with different exceptional situations (internal faults)
  - Compensation handler
    - Undoing persisted effects of already completed activities
  - Termination handler
    - Dealing with forced scope termination (external faults)



Process Instance Lifecycle

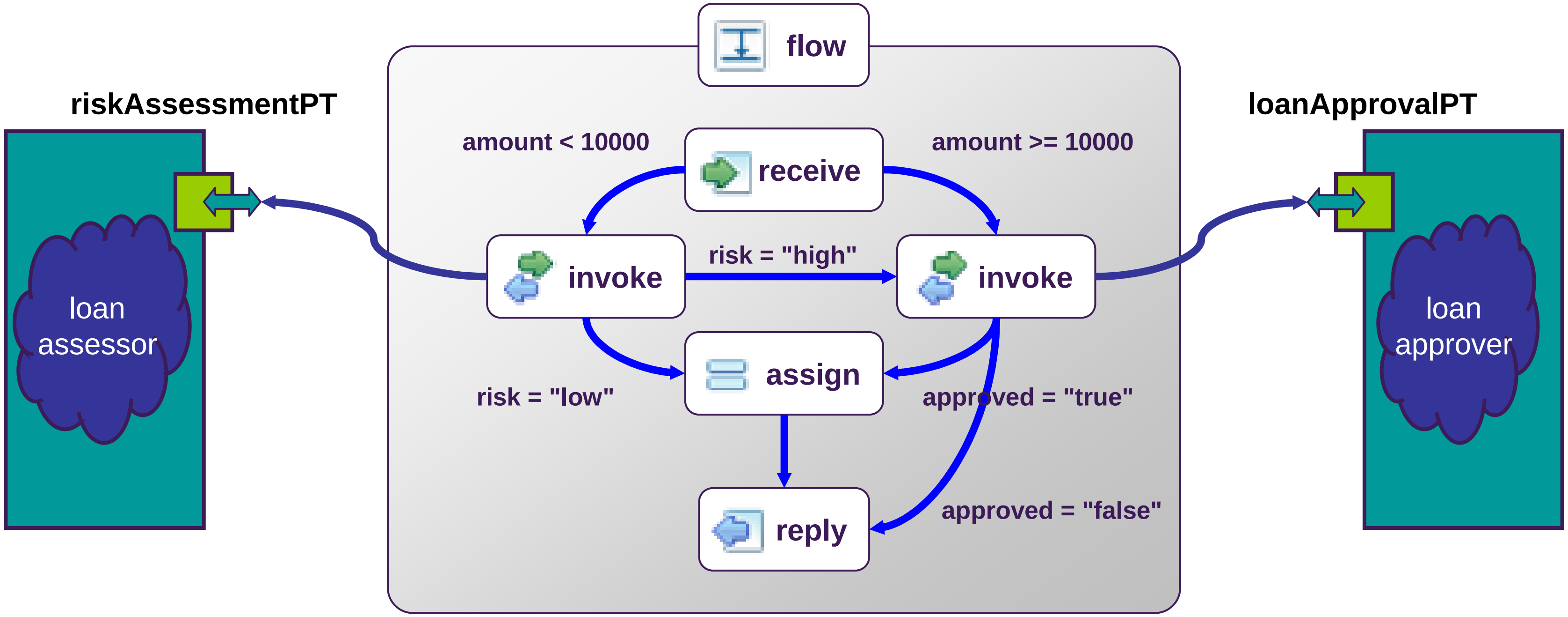
- Business processes defined in BPEL represent stateful Web services
  - When a process is started, a new instance is created
  - The creation and destruction of BPEL process instances is by design implicit



Outline

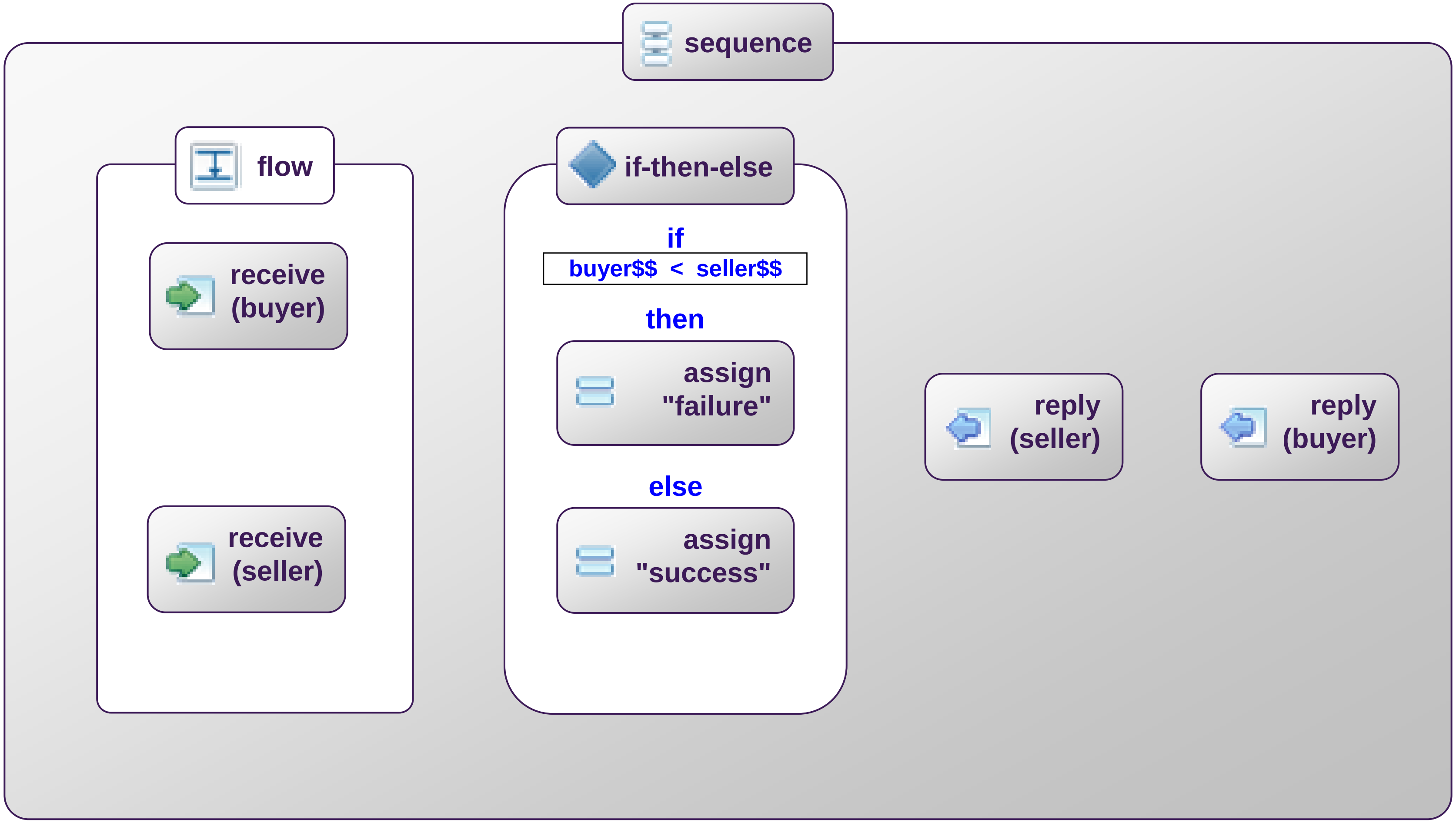
- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support

Graph-Oriented Authoring Style

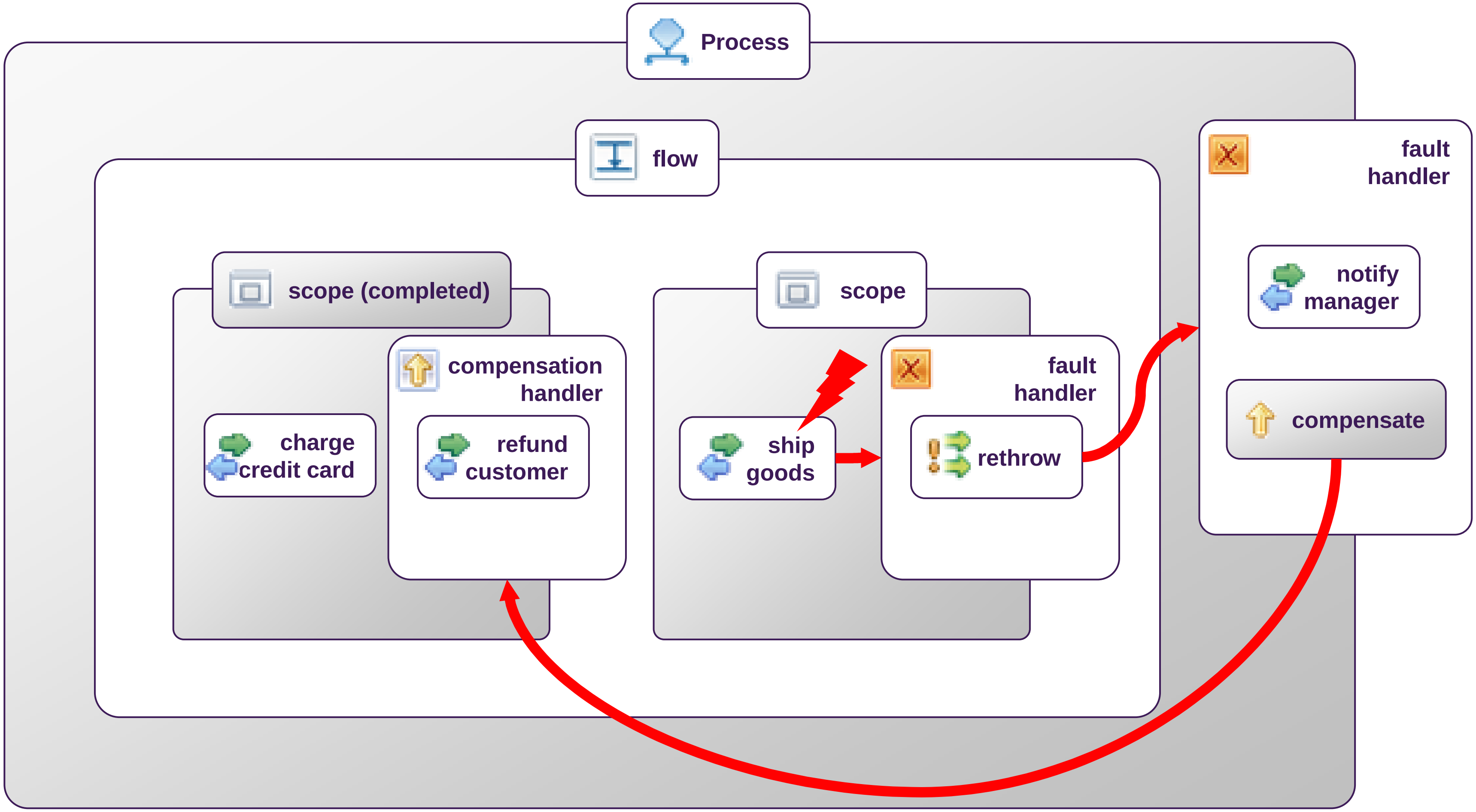


1. A customer asks for a loan, providing name and amount info
2. Two services are involved:
    a) A risk assessor which can approve the loan if the risk is low
    b) A loan approver which checks the name and approves/disapproves the loan
3. The reply is returned to the customer

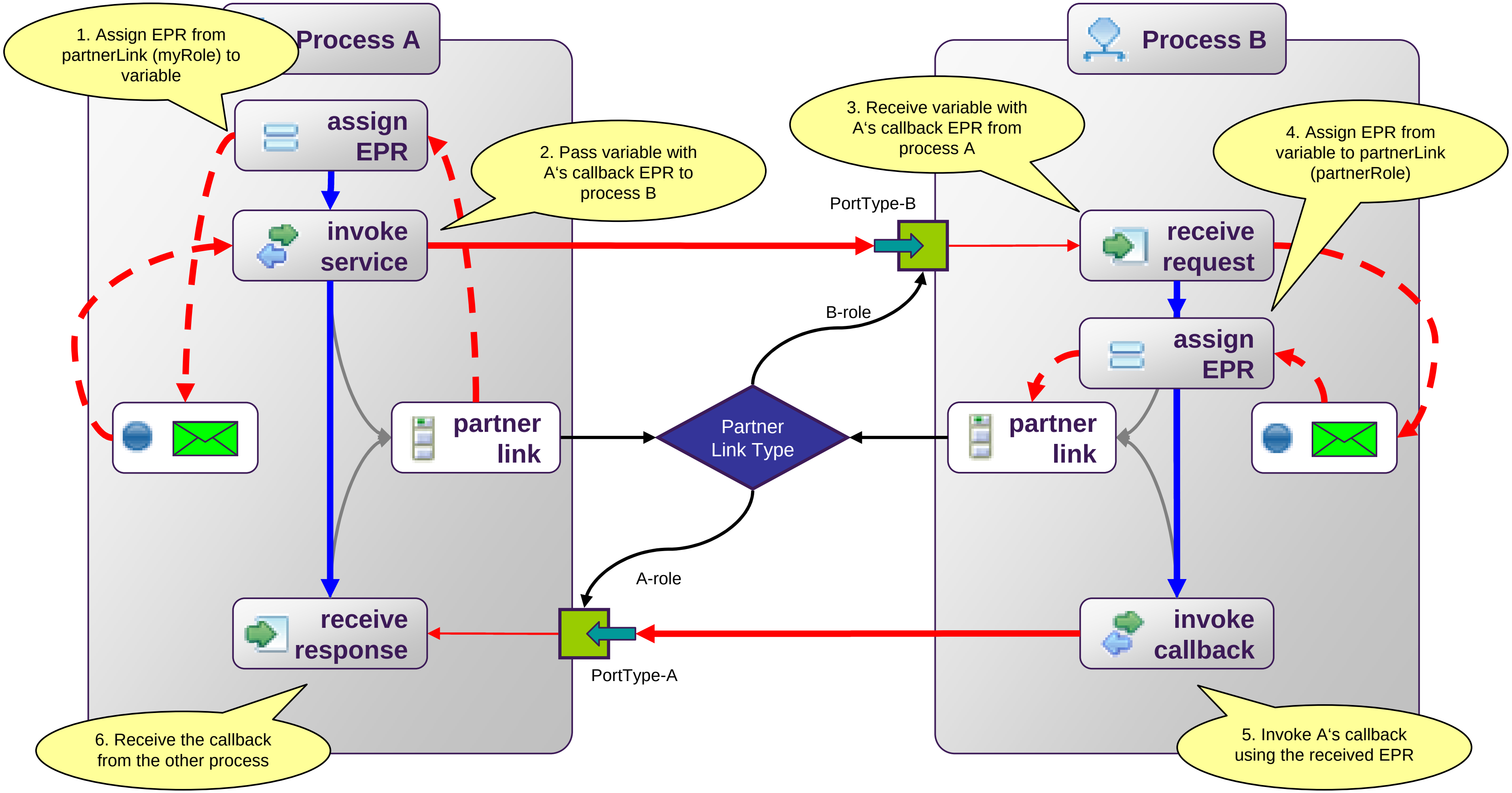
Structured Authoring Style



Fault Handling and Compensation



BPEL Partner Link Assignment



1. Assign EPR from partnerLink (myRole) to variable

2. Pass variable with A's callback EPR to process B

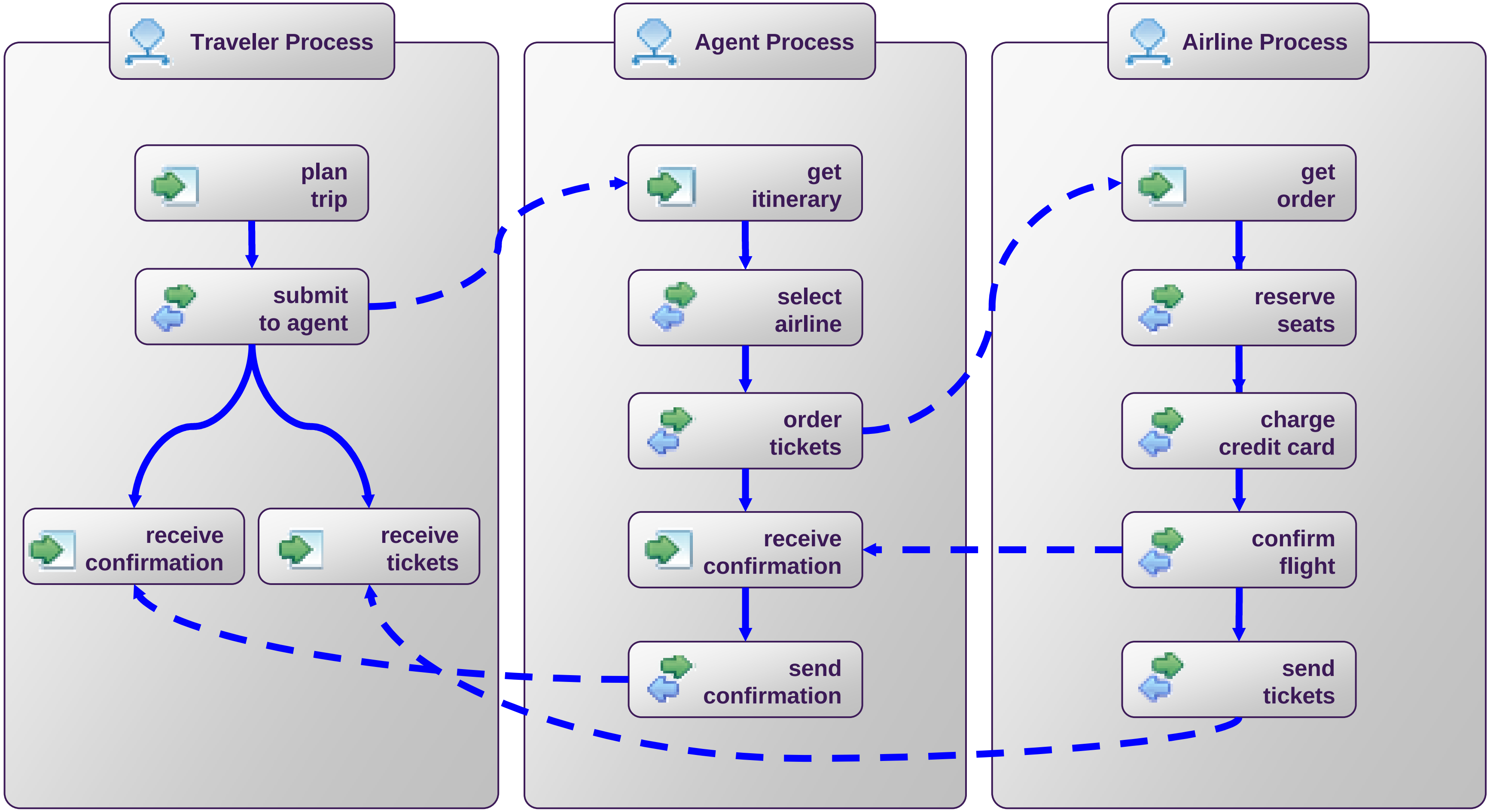
3. Receive variable with A's callback EPR from process A

4. Assign EPR from variable to partnerLink (partnerRole)

5. Invoke A's callback using the received EPR

6. Receive the callback from the other process

Executable Processes View



Outline

- Motivation
- OASIS and WS-BPEL
- Main Concepts
- Examples
- Status and support

WS-BPEL

- Portable, interoperable process model for long running business processes
- Flexible integration of Web services
  - WSDL abstract interfaces alone used to define composition
    - Enables two levels of adaptive behavior
      - Abstract partners can be bound to actual services at runtime
      - The process can choose a protocol for communicating with the service at runtime
  - Services whose data definitions do not match can be composed
    - Data transformations can be inlined in process definition

WS-BPEL Adoption: Products

- Active Endpoints ActiveWebflow Server
- ActiveBPEL Engine (open source)
- bexee BPEL Execution Engine (open source)
- Cape Clear Orchestrator
- FiveSight PXE
- IBM WebSphere Business Integration – Server Foundation 5.1
- IBM WebSphere Process Server 6.0
- OpenLink Virtuoso Universal Server
- OpenStorm ChoreoServer
- Oracle BPEL Process Manager
- Parasoft BPEL Maestro
- SeeBeyond eInsight BPM
- Twister (open source)

WS-BPEL Application Areas

- Business Process Design
- Autonomic Computing
- Grid Computing
- Semantic Web

What's new since BPEL4WS 1.1?

- Activities: if-then-else, repeatUntil, validate, forEach
- Extension activity
- Completion condition in forEach activity
- Variable initialization
- XPath access to variable data

```
$variable[.part]/location
```

- XML schema variables for WS-I compliant doc/lit-style WS interactions
- Locally declared messageExchange for correlating receive and reply activities
- Abstract processes – common base and profiles

WS-BPEL 2.0 To-Do List

- Important open issues
  - – Miscellaneous specification clarifications
  - – Abstract processes
    - Common base (syntax)
    - Profiles (semantics)
      - – Externally observable behavior (as in BPEL4WS 1.1)
      - – Templating

Next Steps & Future Direction

- Human user interactions – BPEL4People (as known from existing workflow engines)
  http://www-128.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/
- Subprocesses
  - Based on a coordination protocol
- Transaction semantics
- Currency with related standards
  - WSDL 2.0, XQuery, etc.

WS-BPEL Resources

- OASIS Technical Committee
  http://www.oasis-open.org
- BPEL4WS 1.1
  http://dev2dev.bea.com/technologies/webservices/BPEL4WS.jsp
  http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
  http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbiz2k2/html/bpel1-1.asp
  http://ifr.sap.com/bpel4ws/
  http://www.siebel.com/bpel
- WS-BPEL 2.0 – latest approved committee draft (September 2005)
  http://www.oasis-open.org/committees/document.php?document_id=14314&wg_abbrev=wsbpel
- Info aggregator sites
  - Wikipedia
    http://en.wikipedia.org/wiki/BPEL
  - BPEL Resource Guide
    http://bpelsource.com
- Numerous books and conference papers
- Analyst reports