



**Hasso
Plattner
Institut**

IT Systems Engineering | Universität Potsdam

Emerging Web Service Technologies

Composite Web Services

Martin Lorenz

Supervisor: Tobias Vogel

Composite Web Services

K.-K. Lau and C.M. Tran. In C. Pautasso and T. Gschwind, editors, *Emerging Web Services Technology*, Volume II, pages 77-95, Birkhauser Verlag, 2008.

<http://www.cs.manchester.ac.uk/~kung-kiu/pub/wewst07.pdf>

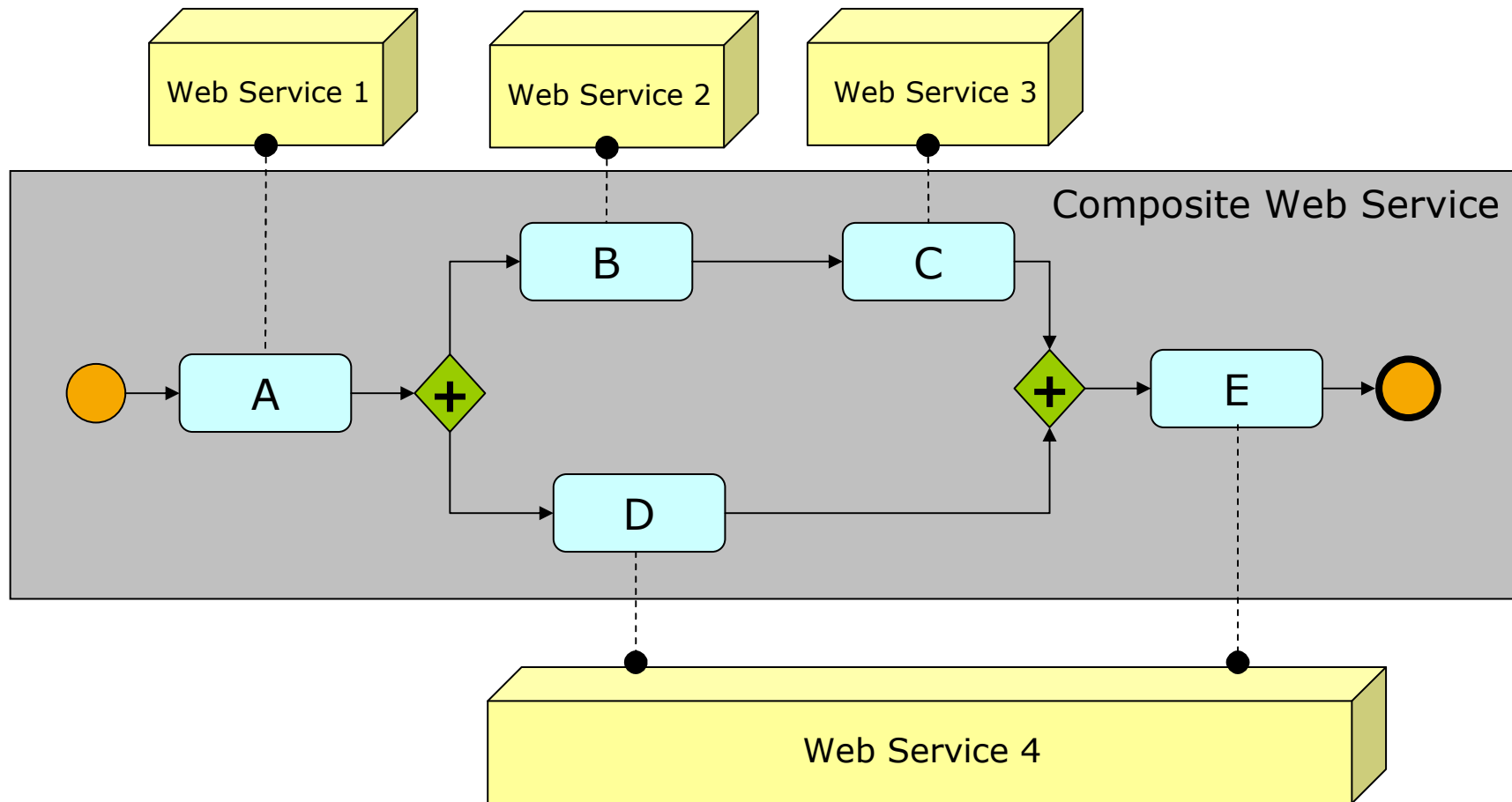
Agenda

3

1. Look at common approach to Web Service Composition
2. Introducing a new view on Composite Web Services
3. Compare the different views
4. Advantages of the new approach
5. Discussion points

Orchestration as Web Service Composition

4



Web Service Orchestration

5

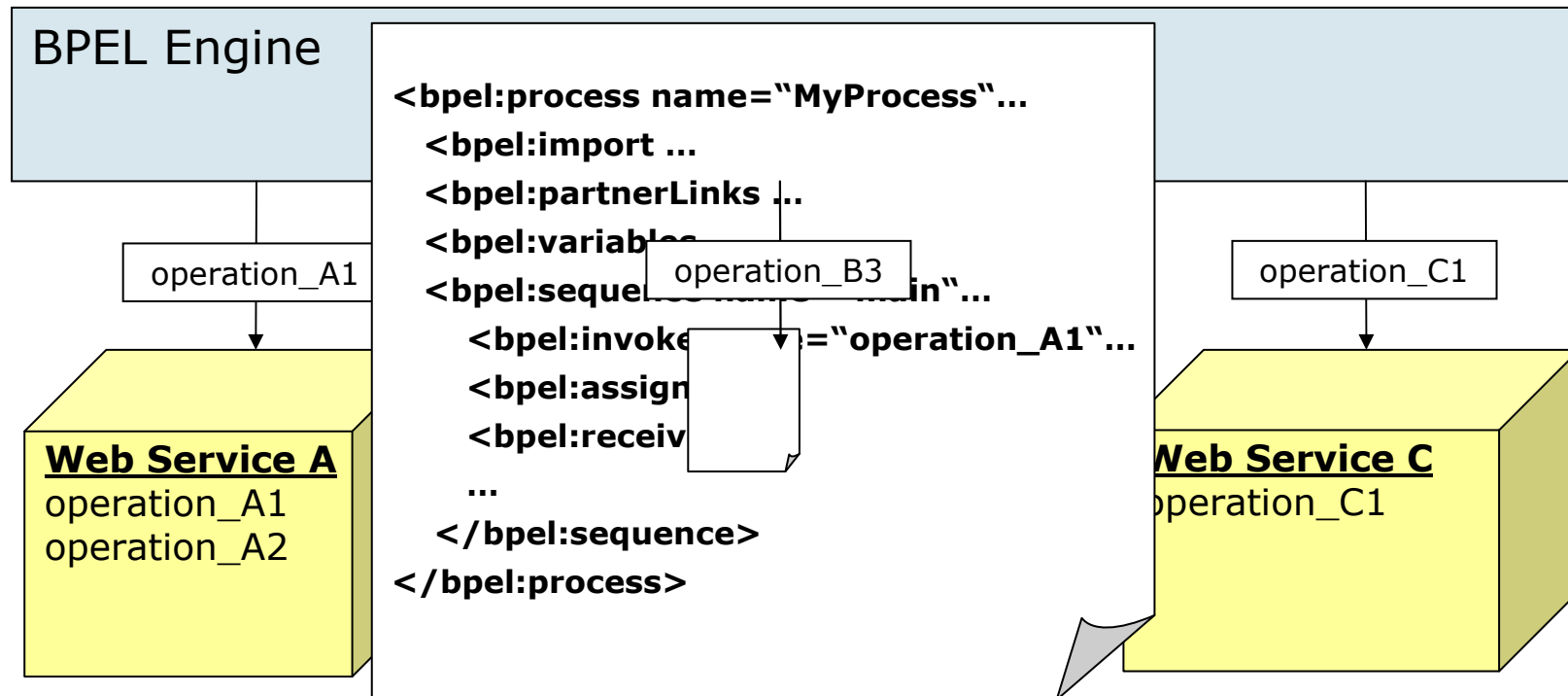


- Web Service Orchestration
 - Sequence of activities
 - Activities invoke selected operation on different web services
 - Control flow connects individual activities
 - Execution engine (BPEL engine) enacts the workflow
 - Exposed as a standard Web Service for invocation



Bottom Line

6



ORC: op x op x ... x op → wf

Agenda

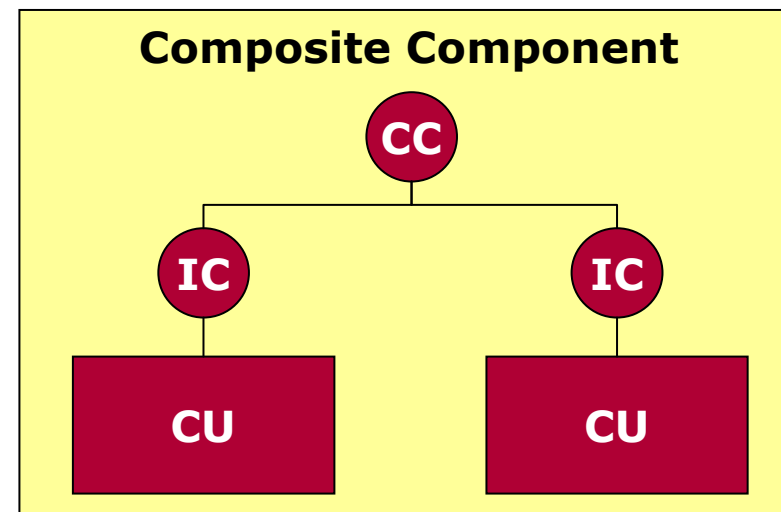
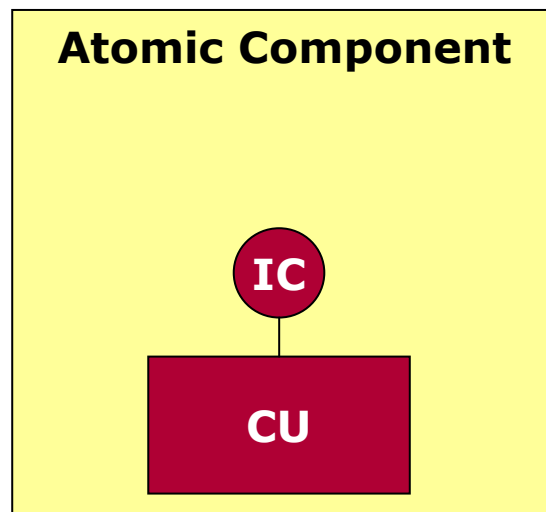
7

1. Look at common approach to Web Service Composition
2. Introducing a new view on Composite Web Services
3. Compare the different views
4. Advantages of the new approach
5. Discussion points

New view on Web Service Composition

8

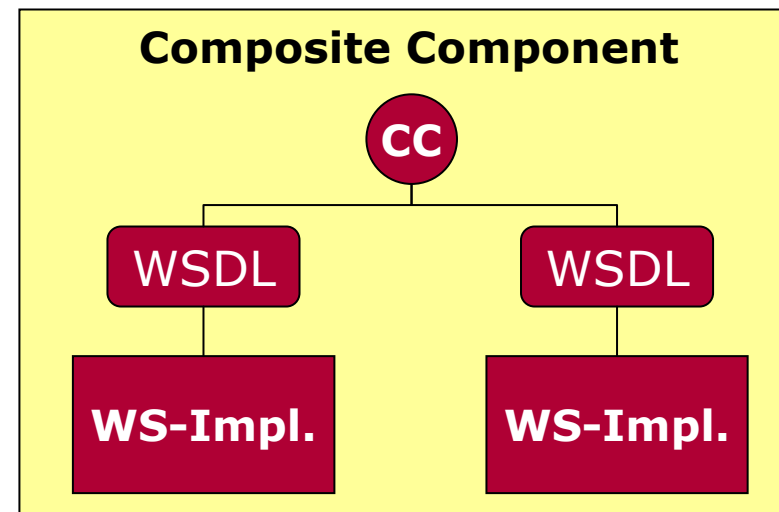
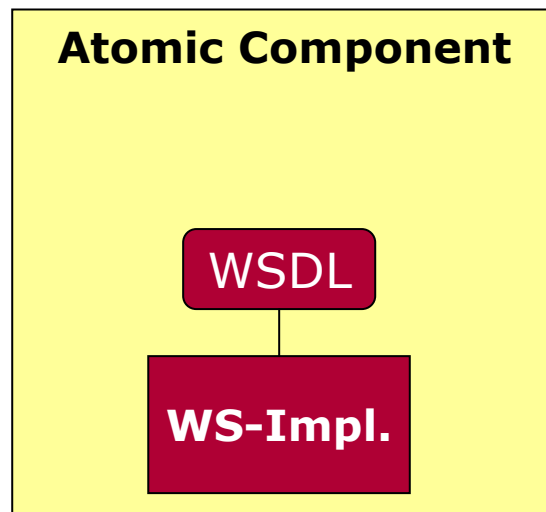
- Lau & Tran developed component model consisting of
 - Computational Unit (CU) → encapsulates computation
 - Exogenous Connectors
 - ◇ Invocation Connectors (IC) → access to methods
 - ◇ Composition Connectors (CC) → encapsulates control flow
 - Compose atomic components into composites
 - Sequence, Pipe and Selector



New view on Web Service Composition

9

- Existing model mapped onto Web Service world
 - Computational Unit → Web Service implementation
 - Invocation Connectors → WSDL
 - Composition Connectors → control flow constructs
- Top-Level CC is the interface to Composite Web Service
 - Defined by file named **Extended WSDL**



New view on Web Service Composition

10

Extended WSDL

- Contains one additional XML element named “workflow”
 - Contains child elements describing detailed workflow structure
- Defines only one operation “invoke”
 - Different signatures depending on connector type

```

<wsdl:message name="invokeRequest">
  <wsdl:part name="condition" type="xsd:string"/> (only selector)
  <wsdl:part name="operations" type="ArrayOfString"/>
  <wsdl:part name="params" type="ArrayOfString"/>
</wsdl:message>
<wsdl:message name="invokeResponse">
  <wsdl:part name="params" type="ArrayOfString"/>
</wsdl:message>

<wsdl:portType name="...">
  <wsdl:operation name="invoke" parameterOrder="cond operations params">
    <wsdl:input message="invokeRequest".. />
    <wsdl:output message="invokeResponse".../>
  </wsdl:operation>
</wsdl:portType>

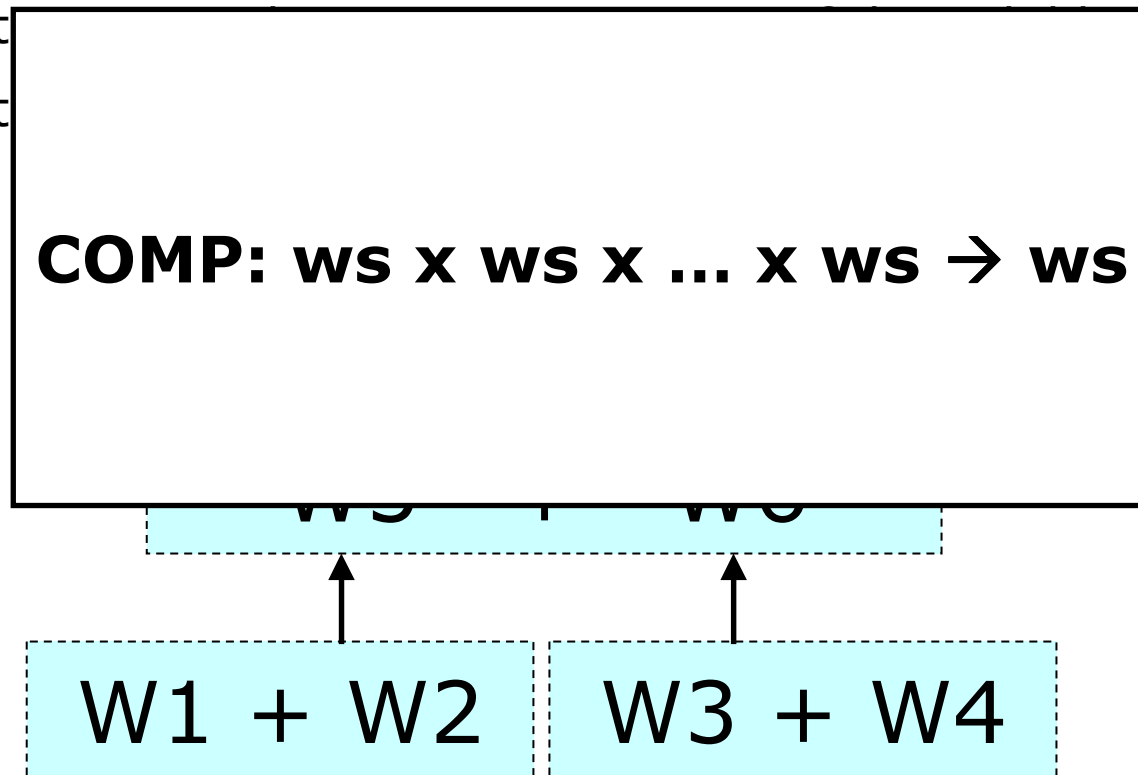
```

New view on Web Service Composition

11

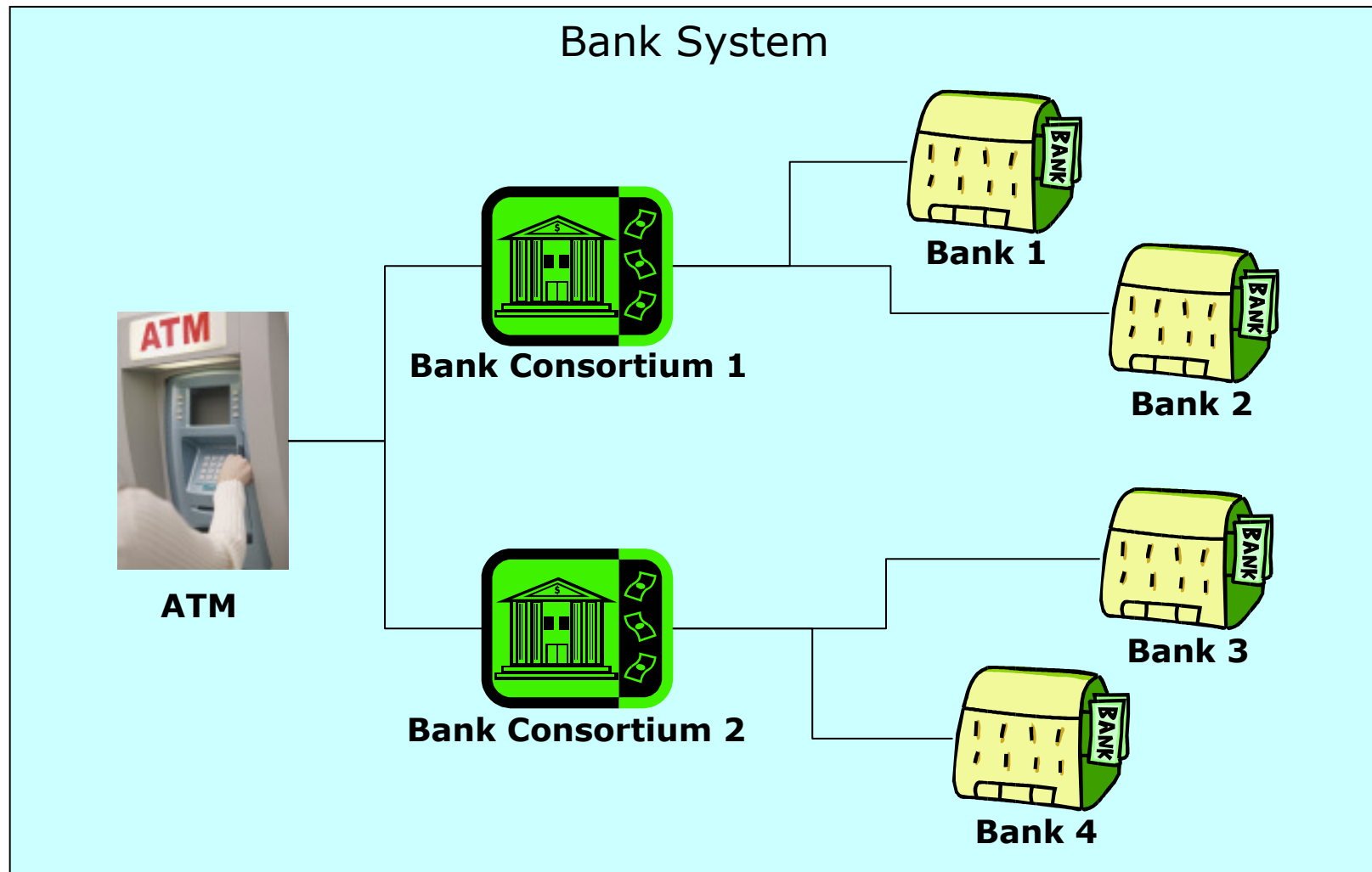
Big Picture

- Hierarchical composition of web services
- Composition
- Composition



Sample Use Case

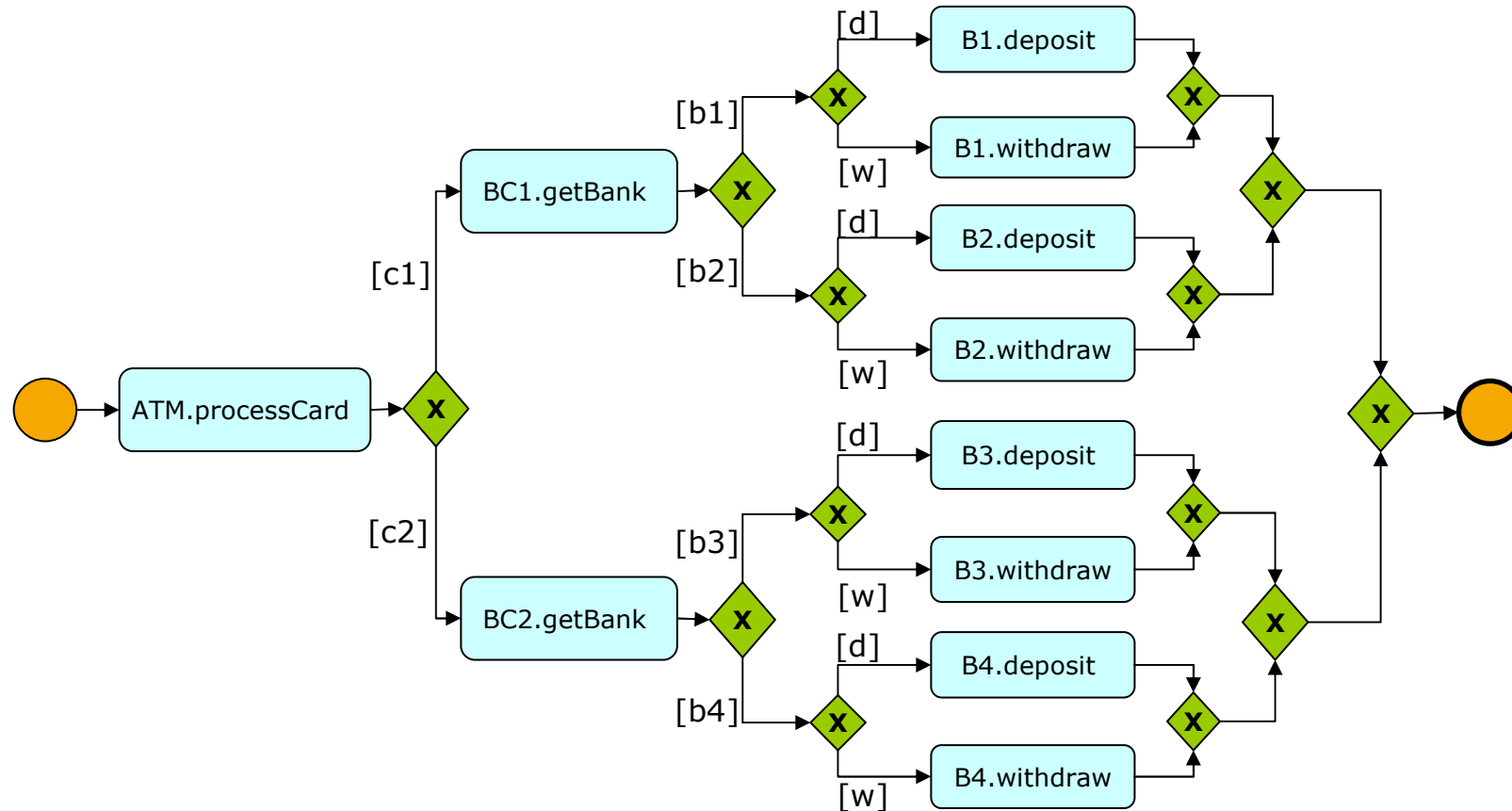
12



Sample Use Case

13

Process Instance: Customer of Bank 1 wants to withdraw 200€



Sample Use Case

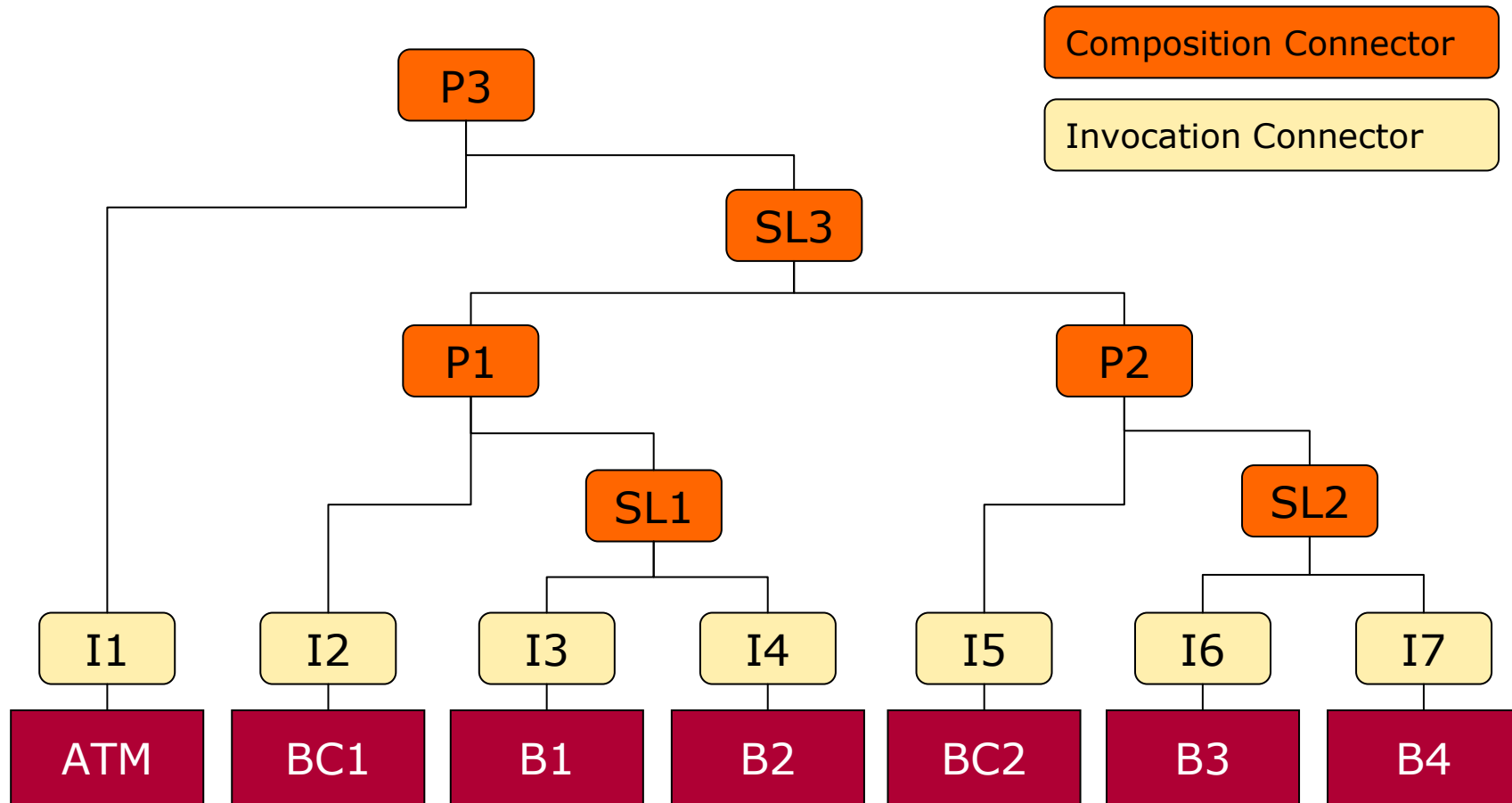
14

- Problems:
 - potentially very large process
 - difficult to manage and to adapt
 - contains redundancies and repetitions
 - duplicated sub-workflows
 - problem adding new operation invocations
 - requires complete new orchestration

Sample Use Case

15

Process flow using alternative approach



Sample Use Case

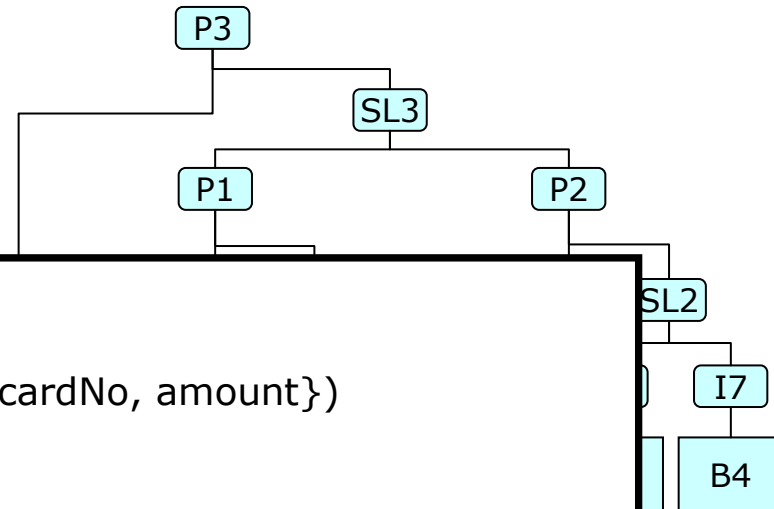
16

Process Instance: Customer of Bank 1 wants to withdraw 200€

```

<workflow>
  <pipe>
    <set name=„ATM“>
      <op name=„procCard“>...</op>
    </set>
    <choice>

```



Client call (withdraw):

invoke({"procCard", "getBank", "withdraw"}, {cardNo, amount})

Client call (deposit):

invoke({"procCard", "getBank", "deposit"}, {cardNo, amount})

```

      <op name=„deposit“>...</op>
    </set>
  </choice>
</pipe>
</workflow>

```

```

...
</workflow>

```


Agenda

17

1. Look at common approaches to Web Service Composition
2. Introducing a new view on Composite Web Services
3. Compare the different views
4. Advantages of the new approach
5. Discussion points

Comparison

18

Composition	Orchestration
All operations available for invocation	Fixed subset of operations available
Compose entire web services to bigger service	Create workflow from individual web service operations
Strictly hierarchical	No hierarchy
Realization of several workflows in a single web service	Realization of a single workflow per orchestration
COMP: $ws \times ws \times \dots \times ws \rightarrow ws$	ORC: $op \times op \times \dots \times op \rightarrow wf$

Agenda

19

1. Look at common approaches to Web Service Composition
2. Introducing a new view on Composite Web Services
3. Compare the different views
4. Advantages of the new approach
5. Discussion points

Advantages of the new approach

20

- All operations available
- Realization of many different workflows instead of a single one
- Developers can focus on building up structure
 - Clients can choose operations based on their business logic
- Low maintenance costs
 - No effort in keeping multiple separate workflows working
- Hierarchical construction of composite services
 - Services on lower level can be used other applications
- Composite is standard WSDL Web Service
 - Can be used in orchestrations

Agenda

21

1. Look at common approaches to Web Service Composition
2. Introducing a new view on Composite Web Services
3. Compare the different views
4. Advantages of the new approach
5. Discussion points

Discussion points

22

- Are all operations of all Services needed?
 - Web Services with large number of operations

Conclusion:

Approach has advantages for certain scenarios. Common approaches like orchestration still cover most of industries requirements.

- No active support by current developer tools