WEBSYDIAN™

Web Services and SOA with CA Plex and Websydian TransacXML

WEBSYDIAN™

Speakers

- Søren Madsen
  Chief Consultant, Soft Design A/S

- Anne-Marie Arnvig
  Communications Manager, Websydian A/S

WEBSYDIAN™

Agenda

- SOA vs. web services
- What is a service?
- What is Service Oriented Architecture?
- The SOA Benefits
- SOA in the Real World
- Services and interfaces
- Going SOA with Plex and Websydian: demo of Websydian TransacXML
- Questions

SOA vs. web services

You can have SOA without doing web services, and you can do web services without having SOA.

SOA is more a methodology than a particular technology. Mindset and discipline are essential.

What is a service?

A service consists of one or a series of contact points where some kind of interaction or interchange occurs:

- Ordering a sandwich in a sandwich shop
- Calling your local tax authorities for instructions
- Retrieving data on a web site
- …

A service is a way of organizing or structuring a series of action processes/interchanges in order to lead the service receiver to the service goal.

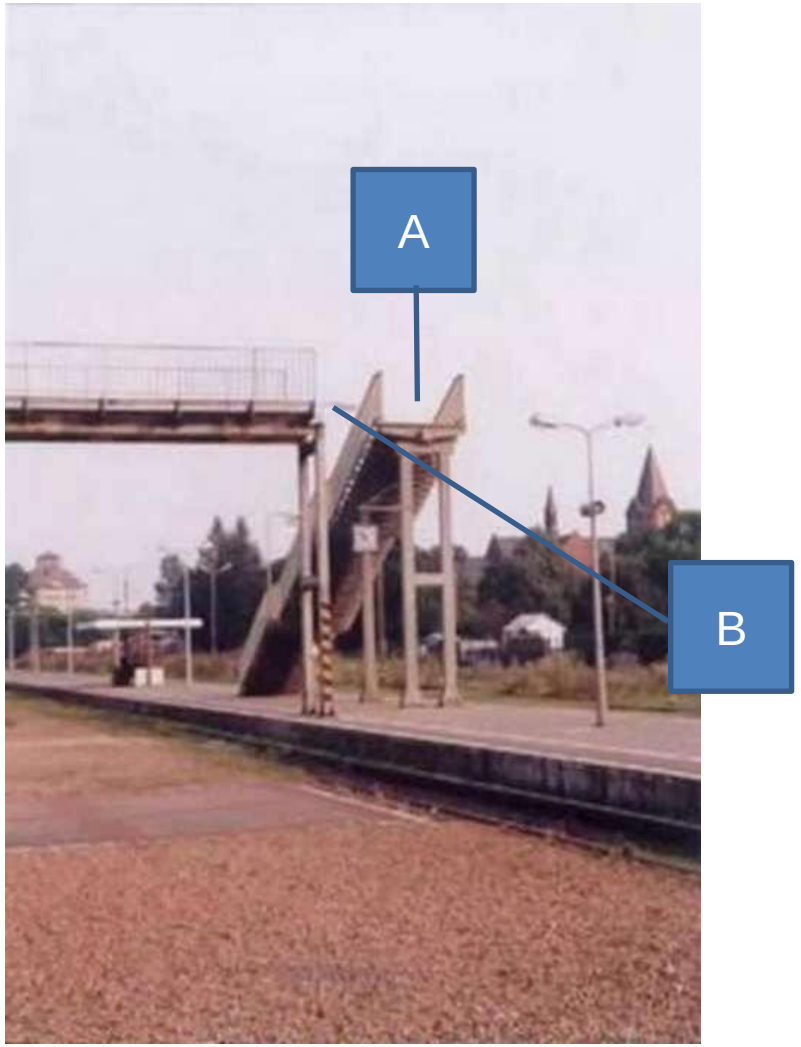
The SOA idea

Each interaction process should interface to the next to finally reach the goal.

If we do it right, we can reuse the processes, but only if the interface contract is in place and fulfilled.

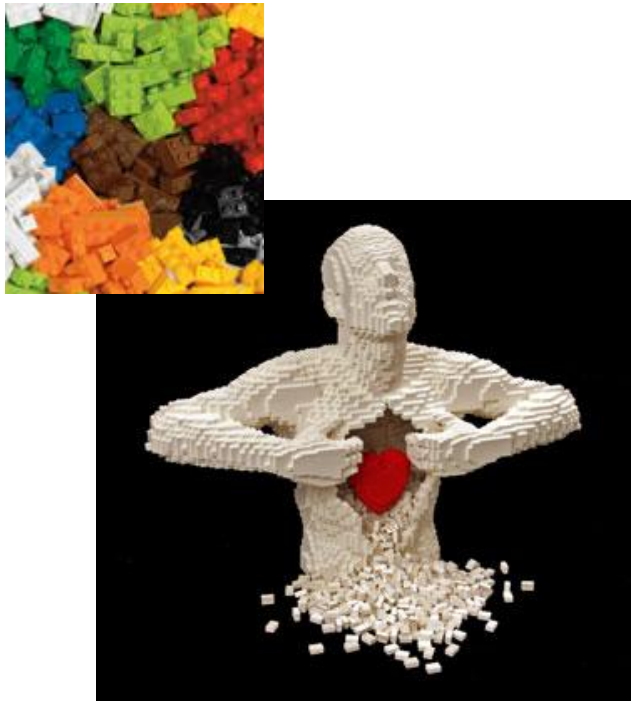
How not to

Both elements fulfill their individual purpose, but the interface contract between A and B is not in place or not fulfilled.



Building blocks with standard interface

- The general SOA idea is to work smarter by constructing systems as sets of building blocks that can all be fitted with each other to create new facilities – one fits all.
- The building blocks are services.
- Built correctly they are so loosely coupled that they can be shared and used for other applications.



WEBSYDIAN™

SOA Benefits

- Code reuse
- Parallel development
- Focused developer roles
- Platform independence
- Greater testability
- Better scalability
- Higher availability

SOA in the Real World

Can we use the SOA mind set at all in the real world?

We all have applications that are not structured like building blocks, and real life doesn't always permit us to work that disciplined.

Luckily you work with CA Plex

You are much closer to SOA than you might think.

CA Plex developers have a head start compared to other coders. With model based development you can profit from these benefits.

Plex ensures:

- modular separation of business processes
- that processes and directories are kept tidy and can be reused by our colleagues effortlessly
- that you can use the SOA mind-set in Plex calls

Going SOA with CA Plex and Websydian

- Consider how you want to structure actions/interchanges as services with Plex calls or web services.
- Use the general idea of building blocks with web services.
- Start thinking in terms of independency regarding platforms, applications etc. using the web services standards: XML, WSDL etc.

- Extend your possibilities and applications
- Get immediate benefit from the SOA idea

WEBSYDIAN™

A practical approach

Søren Madsen, Soft Design A/S

- Chief Consultant
  - Used CA 2E since 1990
  - Used CA Plex since 1996
  - Used TransacXML since 2001
- Speaker at:
  - CA WORLD and Plex/2E user conferences since 2004
  - COMMON US since 2007

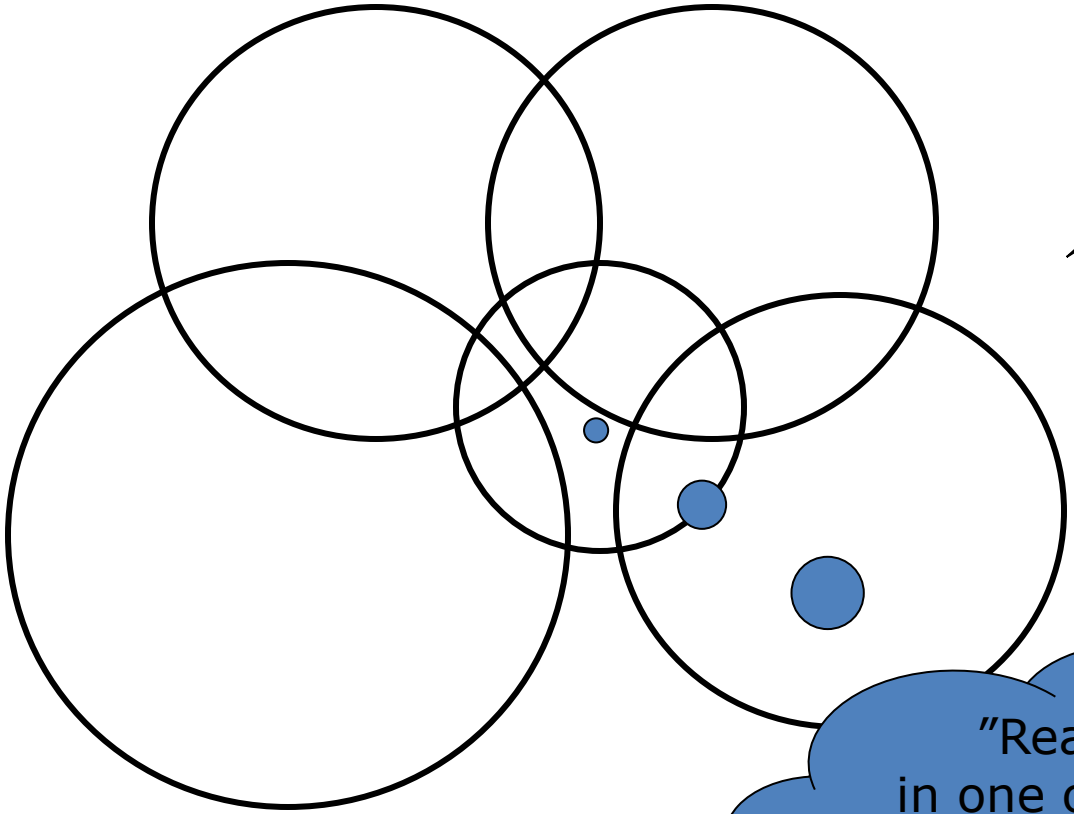
Services

Why all this talk about services?

- You probably already made ”services” yourselves
  - Moduls, API’s, ServiceFunctions, or whatever you call them.

WEBSYDIAN™

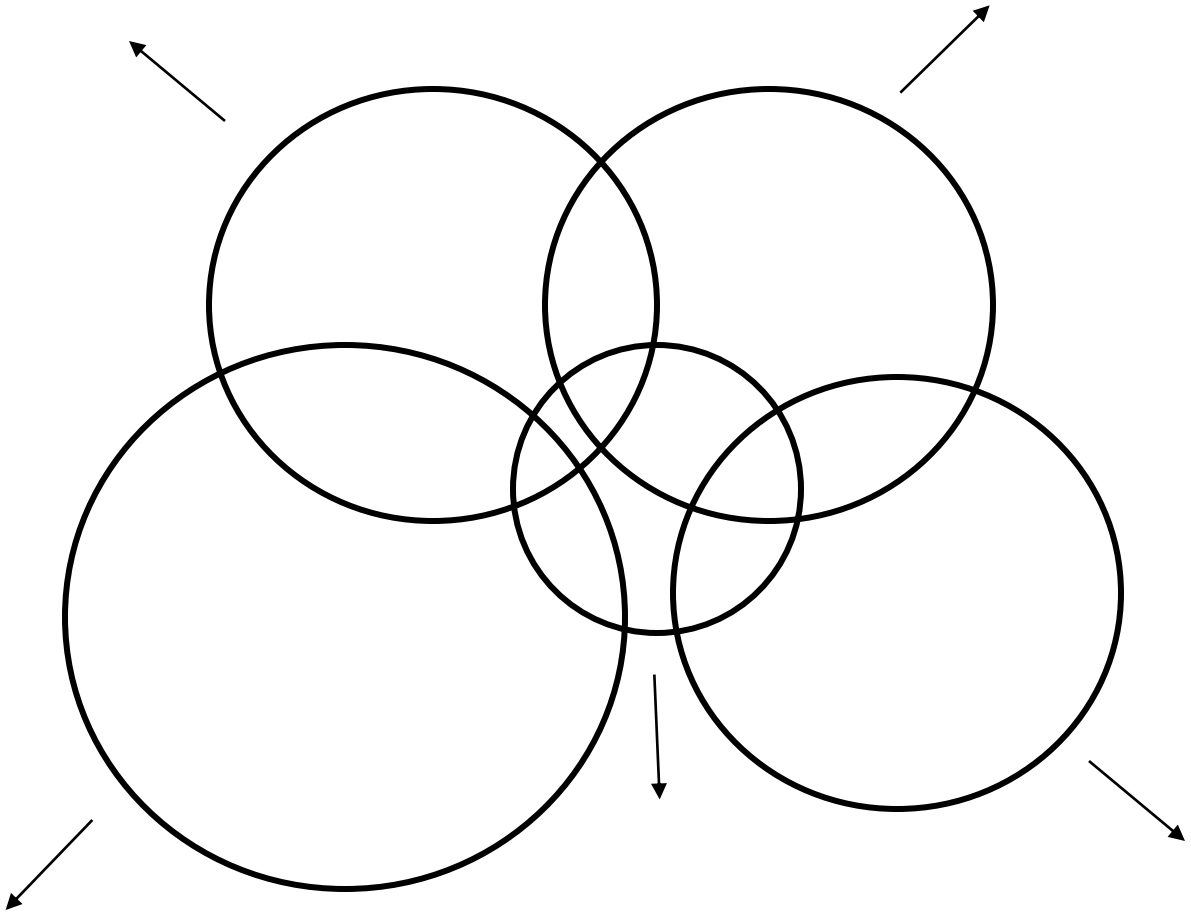
Traditional development and CA Plex systems



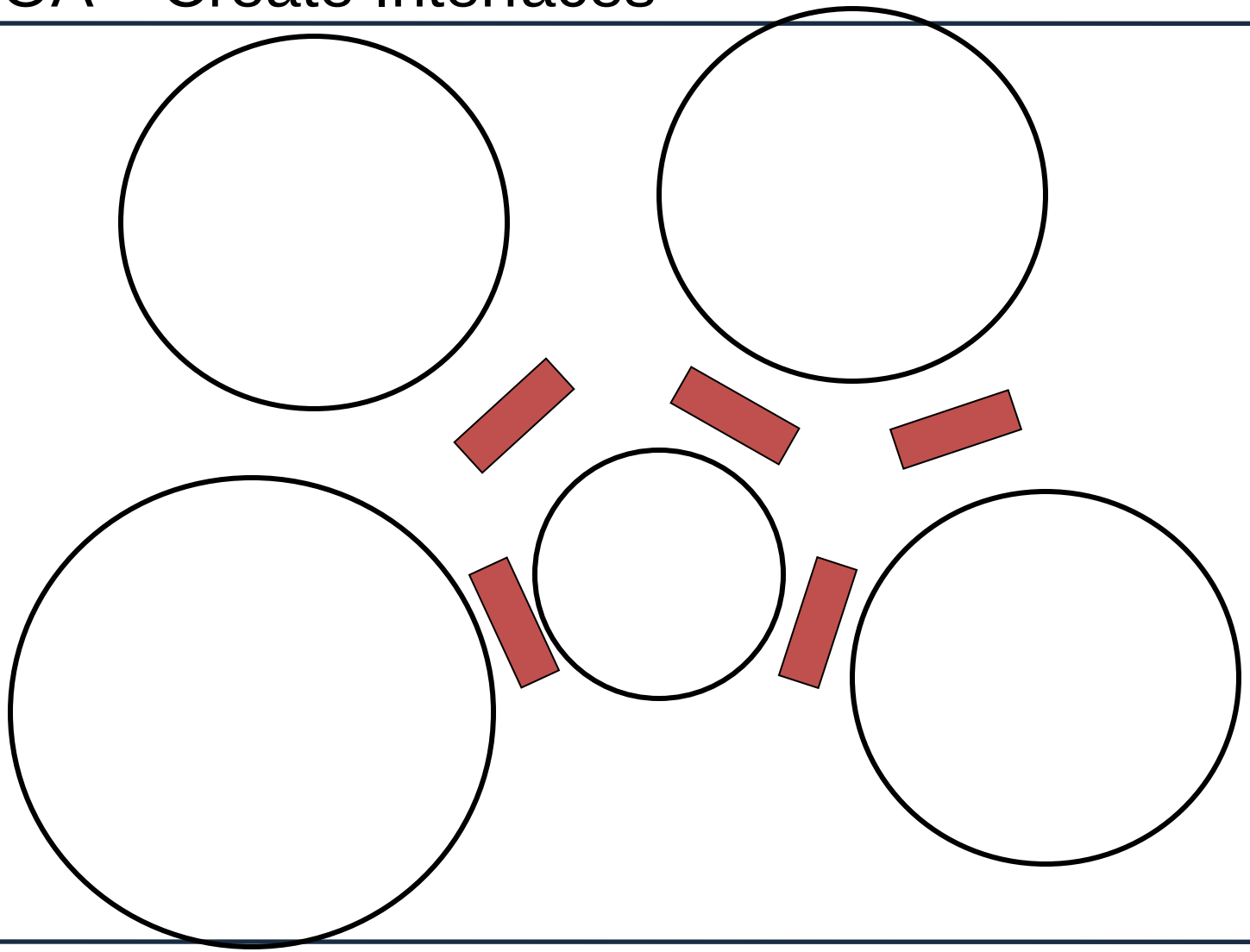
Every circle illustrates a system

”Read an item” in one of our solutions is called by 109 functions => High maintenance costs

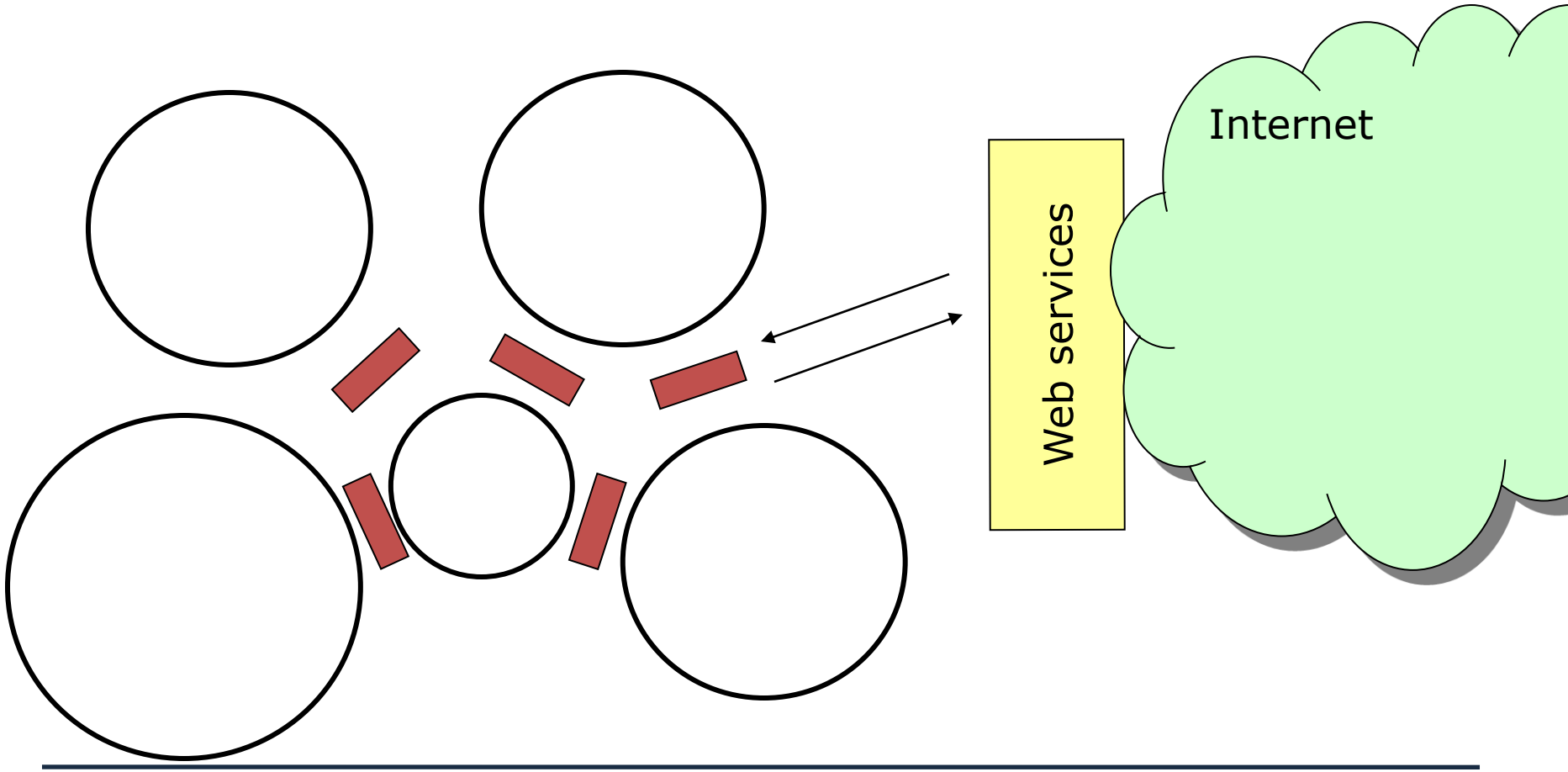
SOA – Separate Our Application Areas



SOA – Create Interfaces



Web services - when relevant



Interfaces

- Have to be stable
  - I.e. a change must NEVER influence a caller.
    - Underlying logic to return an answer can be adjusted as long as the answer remains the same format.
    - If an interface is to be adjusted, a new interface is created
      - So the existing interface is stable.
      - Caller has to change to the new interface if he needs to.

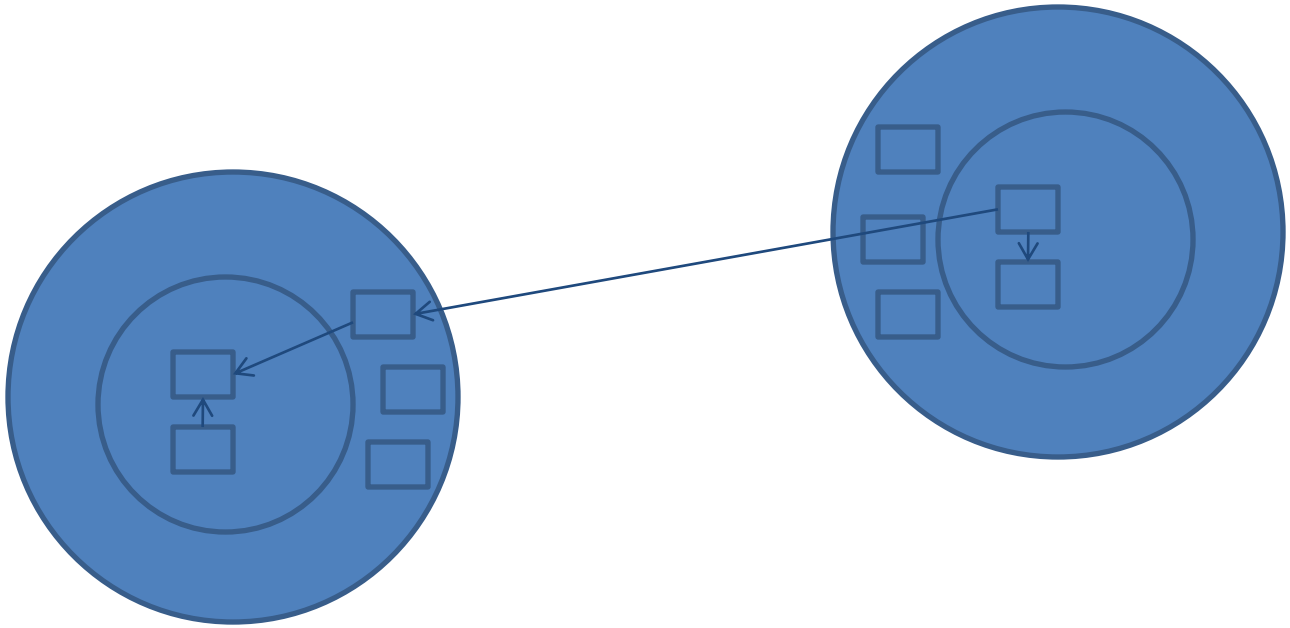
Interfaces

- Have to be ”self-explanatory”
- The name explains the business logic executed.
  - Input/output are the logic information needed to do the job.
  - I have often made interfaces that “do too much”
    - Hard to use since input/output aren’t self-explanatory
    - Hard to test and maintain because you influence a lot of “services”

If we focus on the notion of web services rather than architecture, SOA is great for:

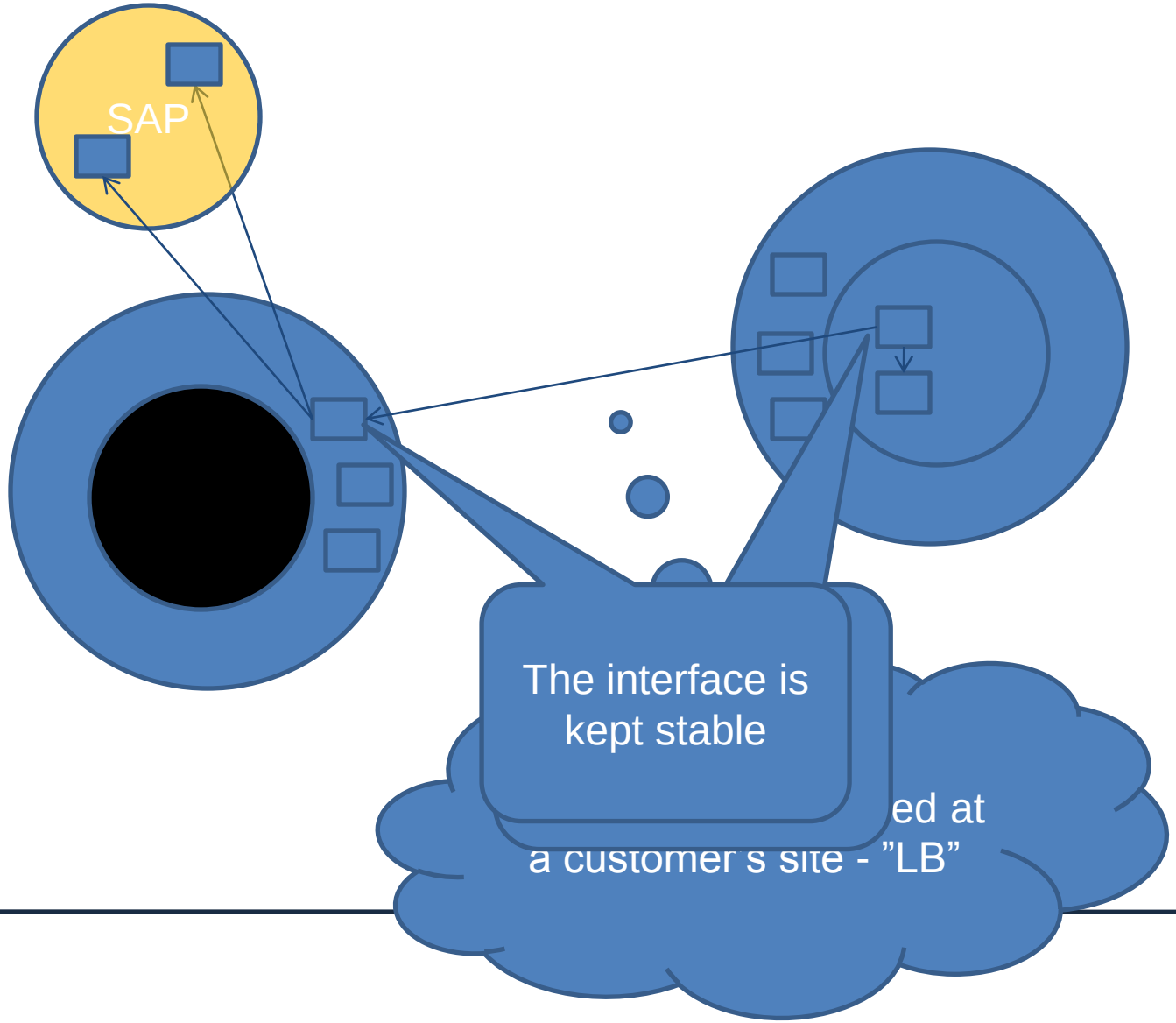
- enabling communication between applications over the Internet
- establishing communication between services in different systems

Loose coupling



WEBSYDIAN™

Loose coupling



WEBSYDIAN™

Demo

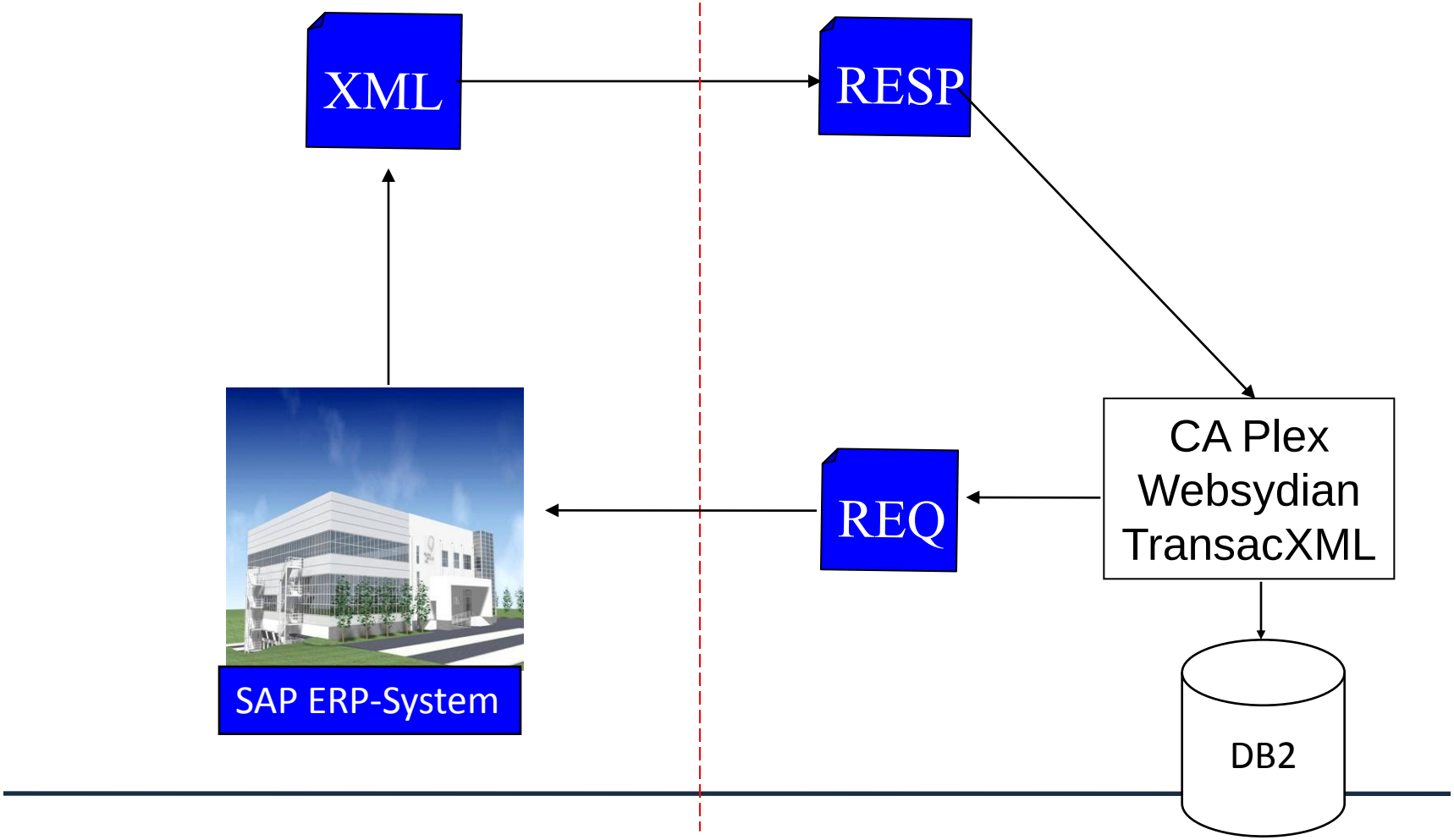
- How to develop the XML part of a web service using CA Plex and Websydian TransacXML

Demo Basics

WEBSYDIAN™

- Management just bought a new ERP system.
- The IT-department is required to keep periferal systems working
- In this case the ERP-system is SAP.
- The operation used in this demo is a Create Customer
- Focus is on creating and receiving the XML – Not the transport layer HTTP, FTP, MAIL .....

Web services – SAP Demo



Demo Challenges

WEBSYDIAN™

- How do I define the XML
- How do I create the XML Document
- How do I send the XML
- How do I receive the XML Data Response
- How do I traverse trough the XML Sheets

- Use TransacXML with WSDL/Schema Import

Live demo

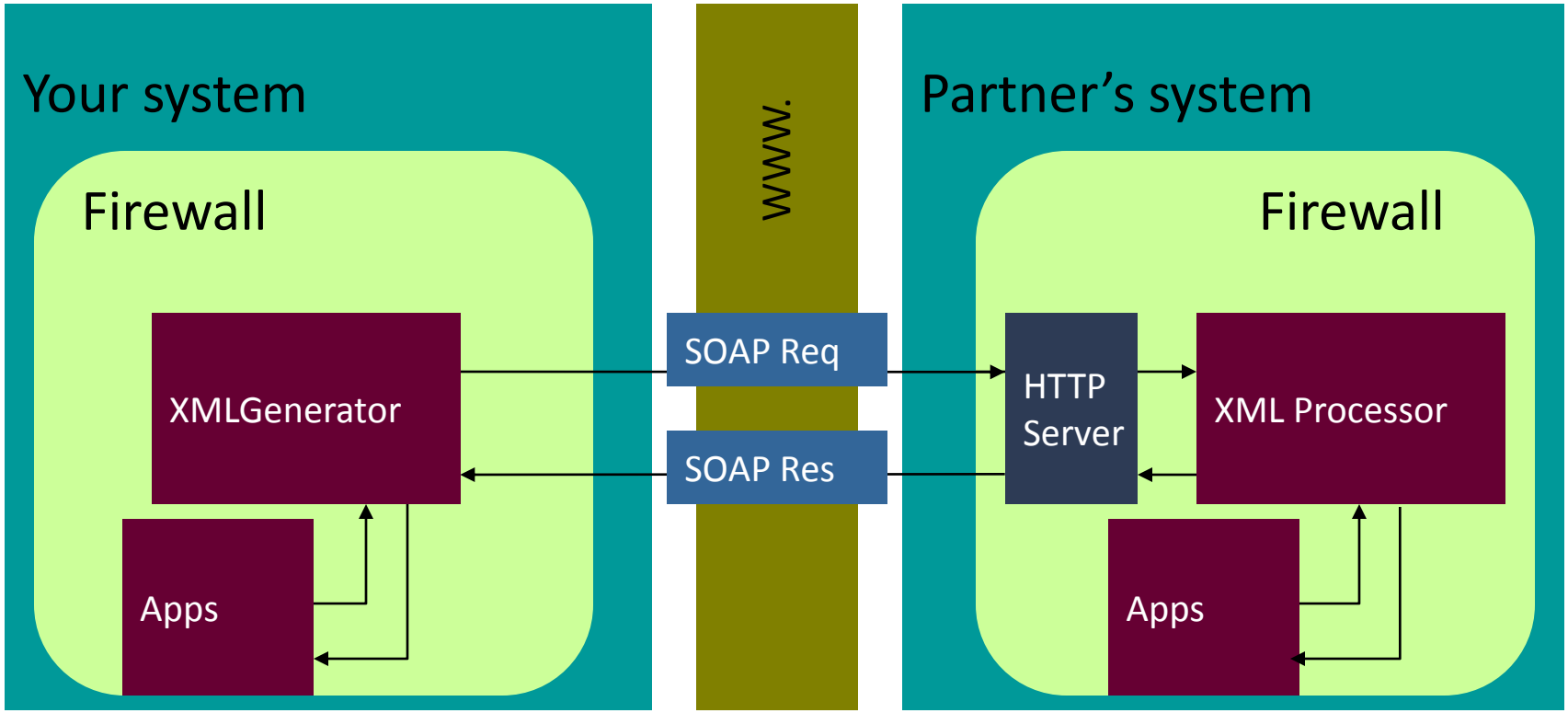
Another example

WEBSYDIAN™

- A partner requires that we interact with them using XML
- In this case the partner has SAP
- The operation used in this demo is a Create Customer
- Focus is on creating and receiving the XML – Not the transport layer HTTP, FTP, MAIL .....

WEBSYDIAN™

Document based web services



Q&A