

SOA, EDA, BPM and CEP are all Complementary

by

David Luckham

Wikipedia

“There is no widely-agreed upon definition of **service-oriented architecture** other than its literal translation that it is an architecture that relies on **service-orientation** as its fundamental design principle.”

What is a *service*?

Think of it as something you want to use, a weather report or a credit rating.

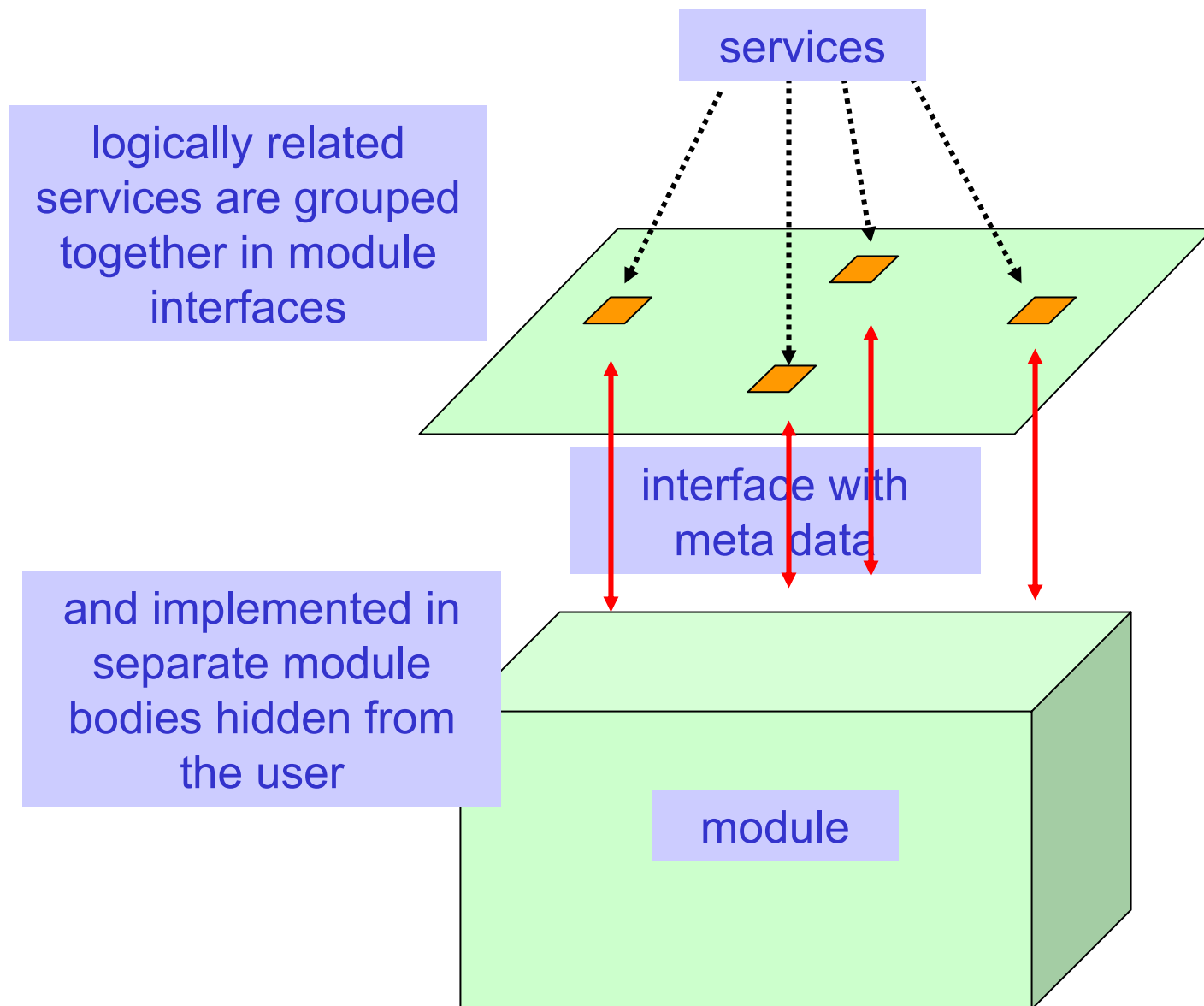
- Send the weather service your zip code and you get back a report for your local area.

- In its purest and oldest form, a service is a function — you call it with data and you get an answer.

Two top level defining concepts:

- ❖ *modularity* (in the computer science sense)
- ❖ *remote access.*

Although I have never had complete confidence in Wikipedia, the entries on SOA are as good a documentation of the state of concept chaos in the SOA implementation world as I have found.



The second principle of SOA is:

- ❖ remote access to services

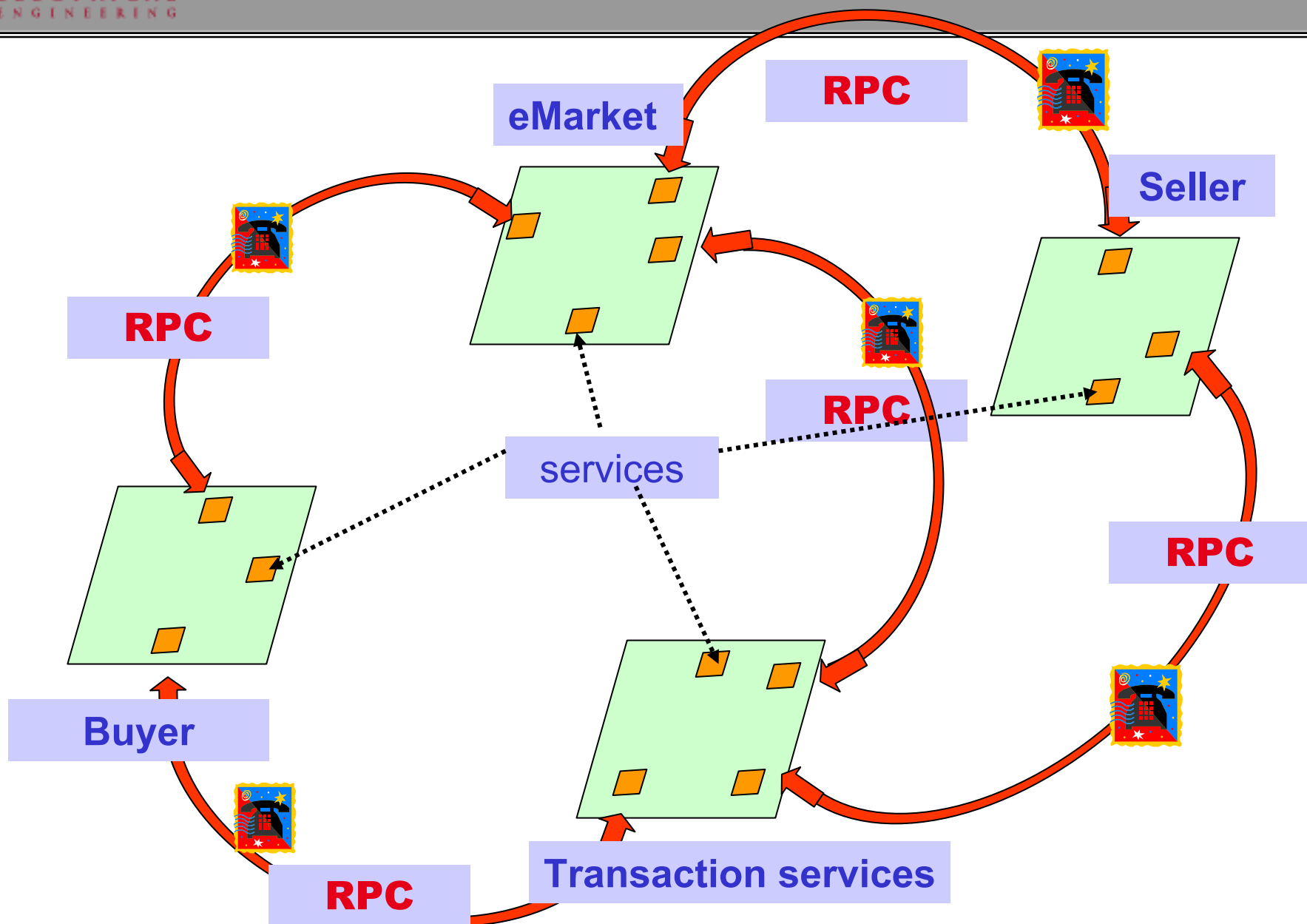
The *remote access* principle of SOA requires that services should operate in a distributed computing environment.

- ❖ SOA must provide users with an ability to access services remotely.

- ❖ Access in traditional SOA [1] was by remote procedure call (RPC) also called Request/Reply (R/R).

[1] By traditional SOA we refer to the state of SOA thinking about the time of the initial release of CORBA in 1991.

Traditional SOA with RPC access



The **up side** of event driven service access is that it is much more versatile and efficient than RPC. It allows all actors – users and services – to multitask.

- ❖ User: “I Need Service”.
- ❖ Airline: “Here is the menu of services for you”.
- ❖ User: “I need human contact. Call back number N”.
- ❖ Airline: “your wait time will be exactly 15 minutes; make sure your call back number is free in 15 minutes, you will receive only one call back event”.

- ❖ The increased versatility has magnified opportunities for everyone – users, services and, of course, **crooks**.
- ❖ **Scalability** has become another issue. We're talking about real time operations with very large numbers of events – e.g., 200,000 business events/sec in stock market operations and the need to make reactive decisions in milliseconds.

In ED-SOA: the management and business intelligence issues are becoming more and more challenging.

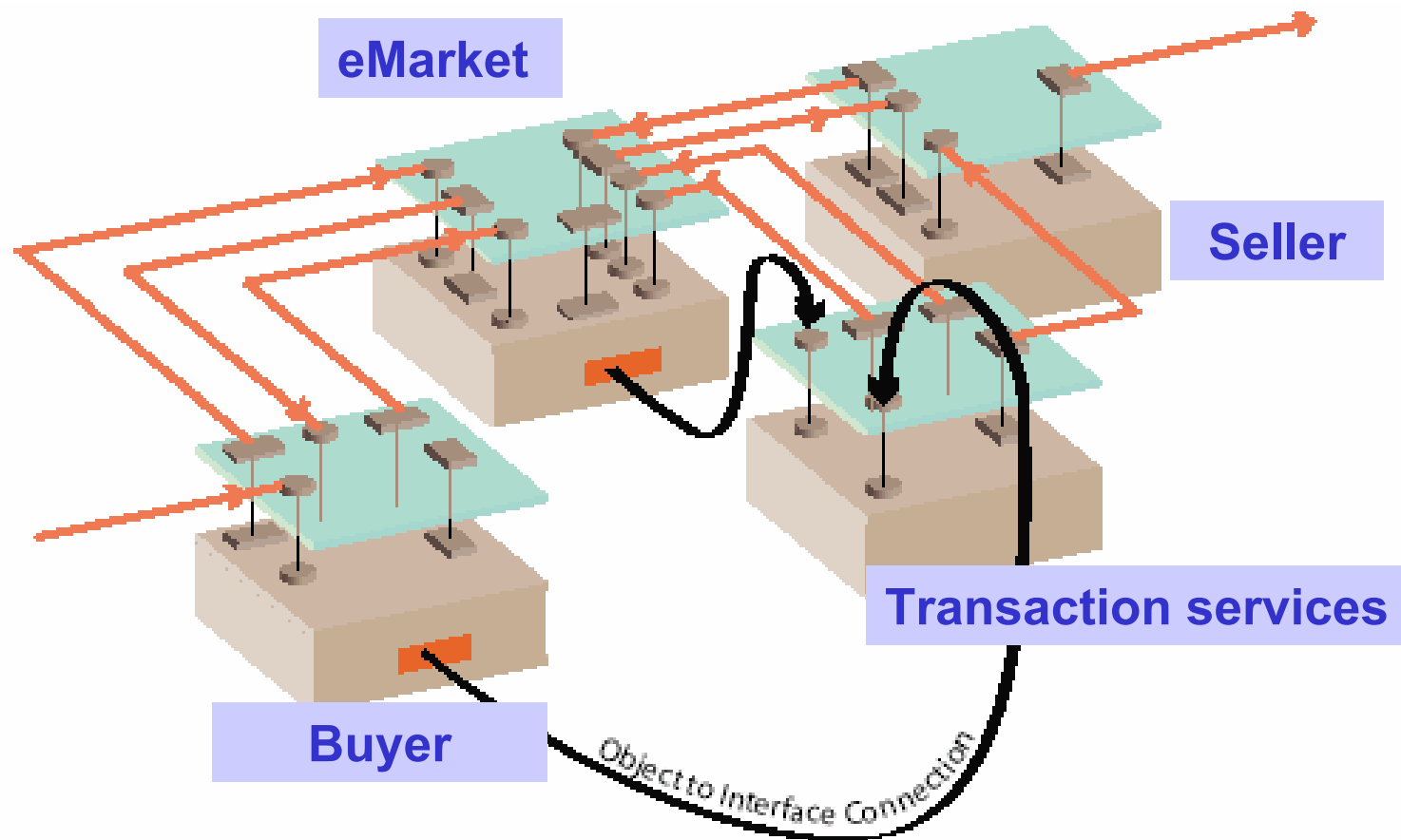
This is where **BPM** and **CEP** come into play

To summarize thus far, SOA and EDA are complementary.

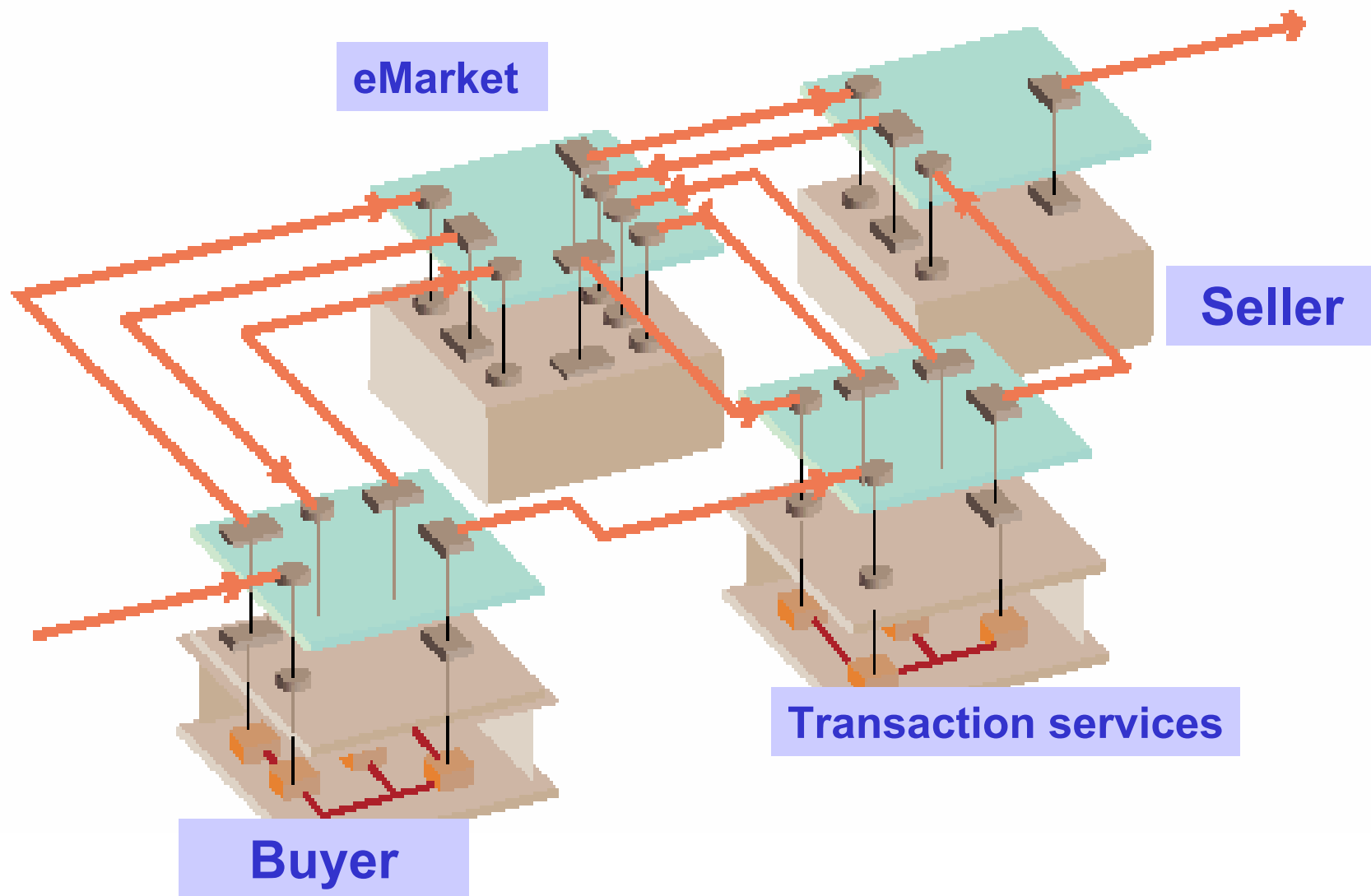
- Event driven communication is paradigm for remote access in SOA, which
- SOA is the best design methodology for EDA.
- all communication is by events and
- all services are reactive event driven processes (i.e., react to input events and produce output events).

- ❖ SOA should support plug & play
1. What are the rules for communication between layers?
 2. Is it practical to apply SOA to all layers of an IT system?

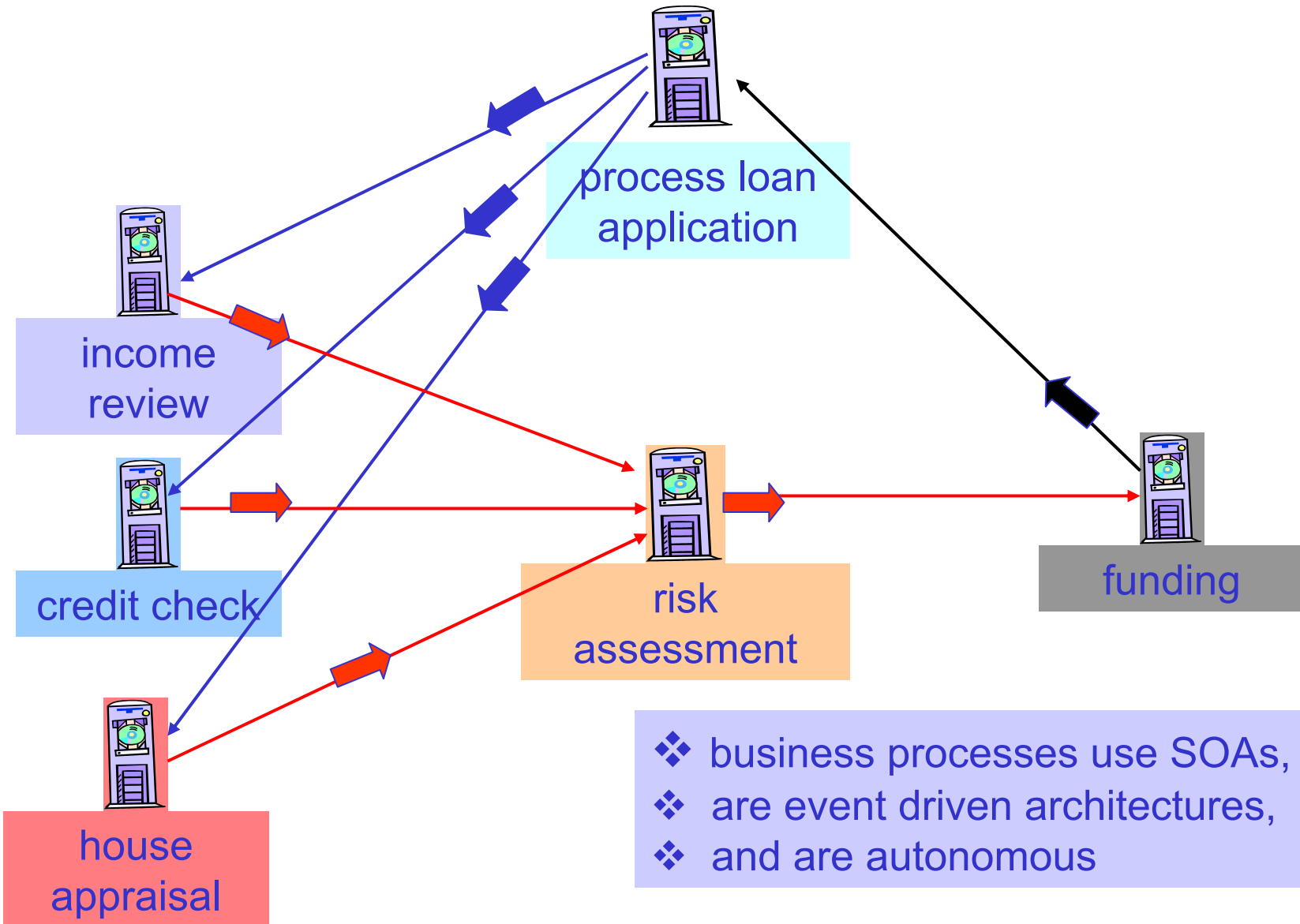
An Object Communication Architecture



- ❖ hierarchical communication rules are violated

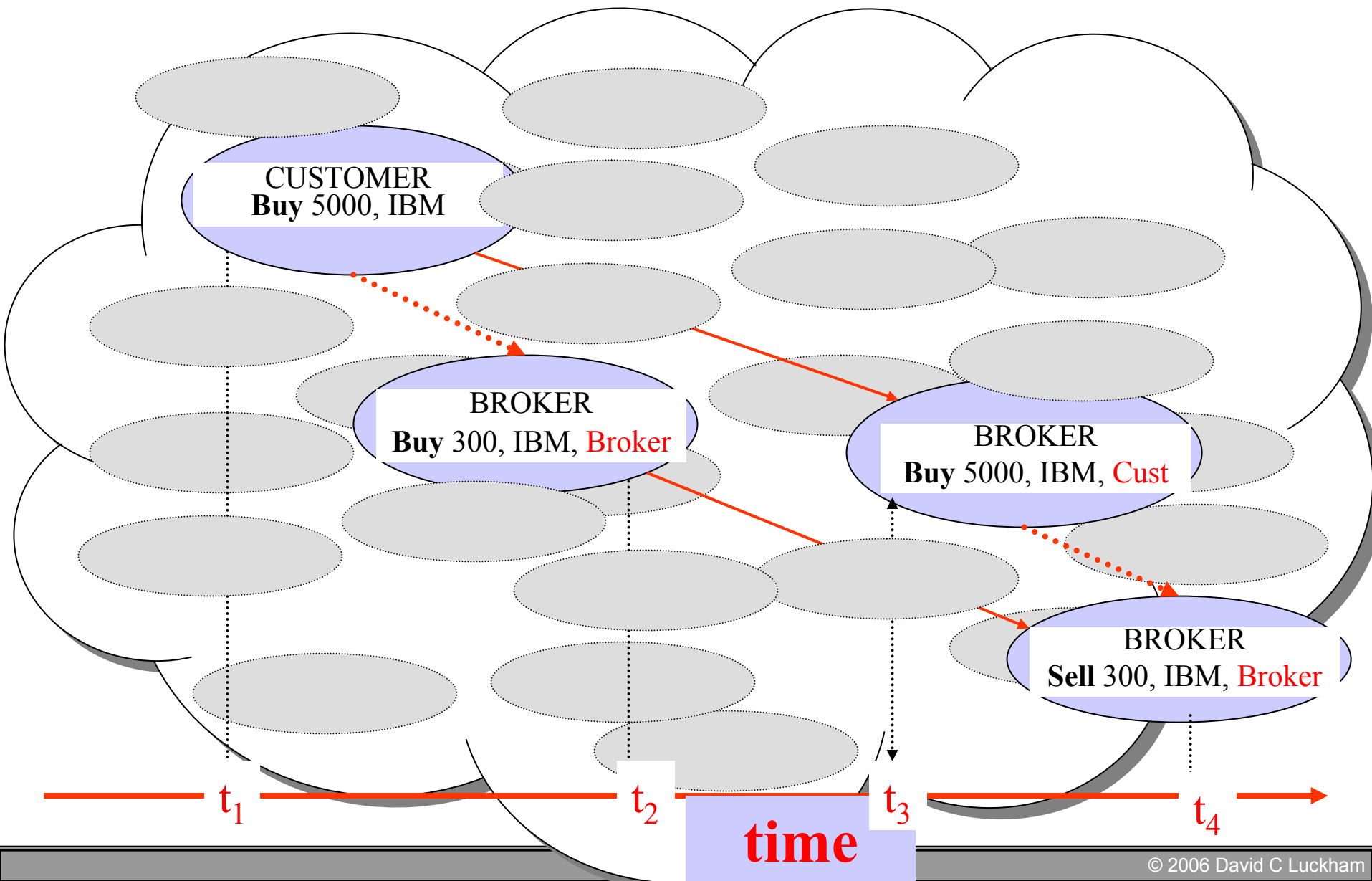


- ❖ ED-SOAs make it easy to build processes.
- ❖ BPM Systems provide a methodology for building ED-SOAs

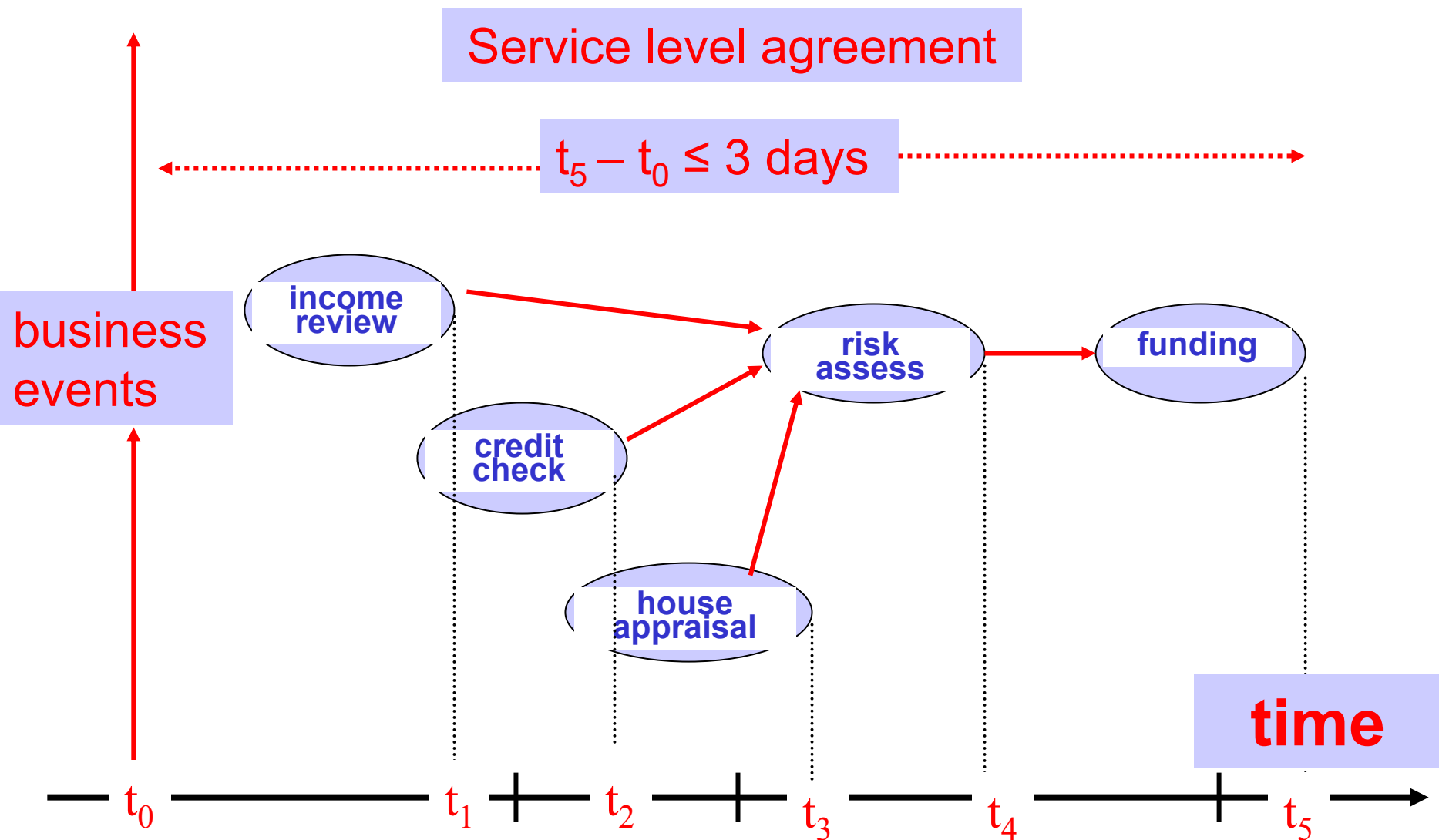


- ❖ business processes use SOAs,
- ❖ are event driven architectures,
- ❖ and are autonomous

- ❖ Application of CEP to monitoring event based systems
- ❖ Application of CEP to autonomous processes



SLA governing process activity



Event patterns to control process activity

if Loan Amount (L) \geq \$1 million then

