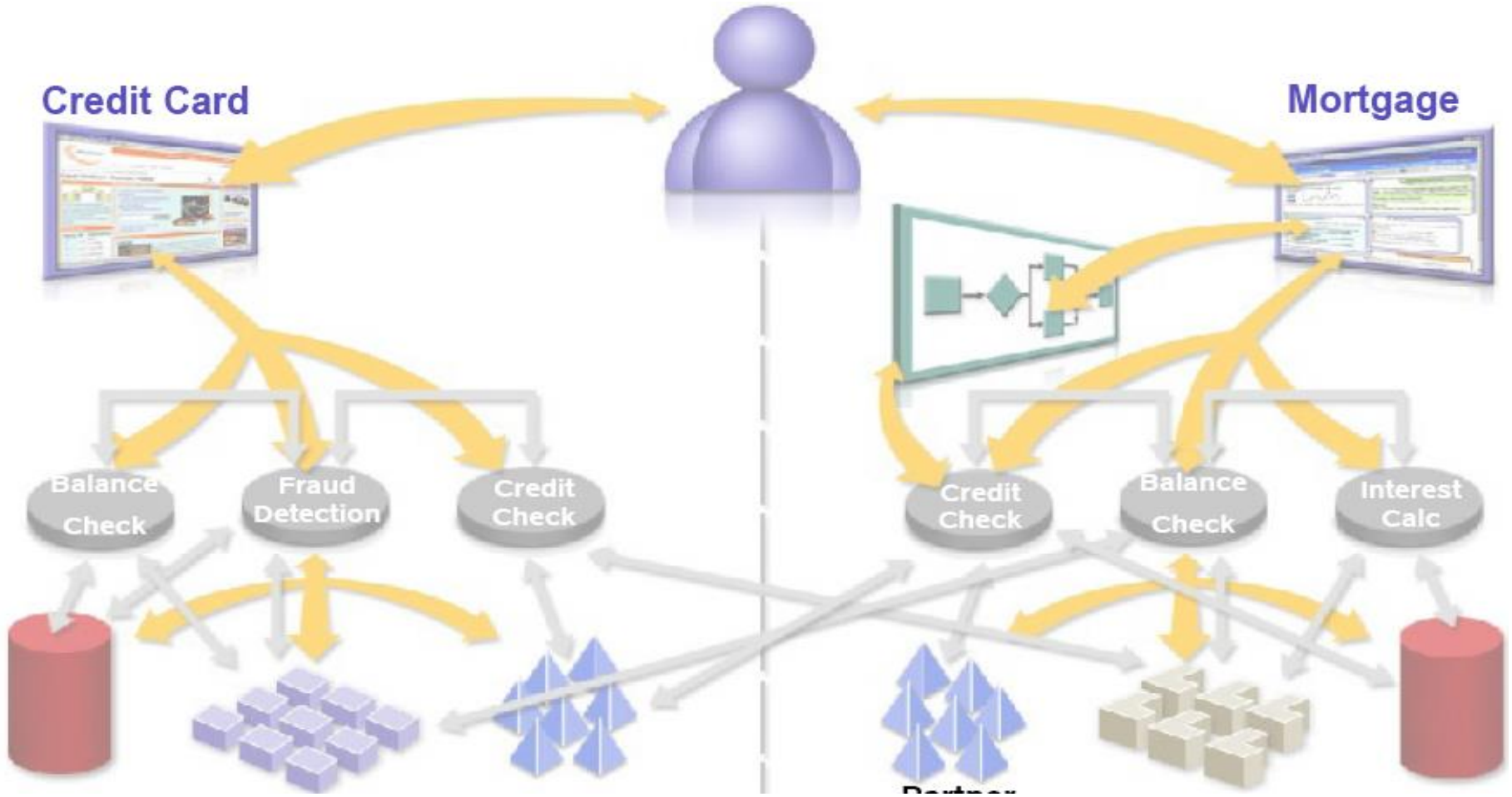
Service Oriented Architecture

Dr. Amit Praseed

A Simple Bank Scenario



Customer Data

Backend System

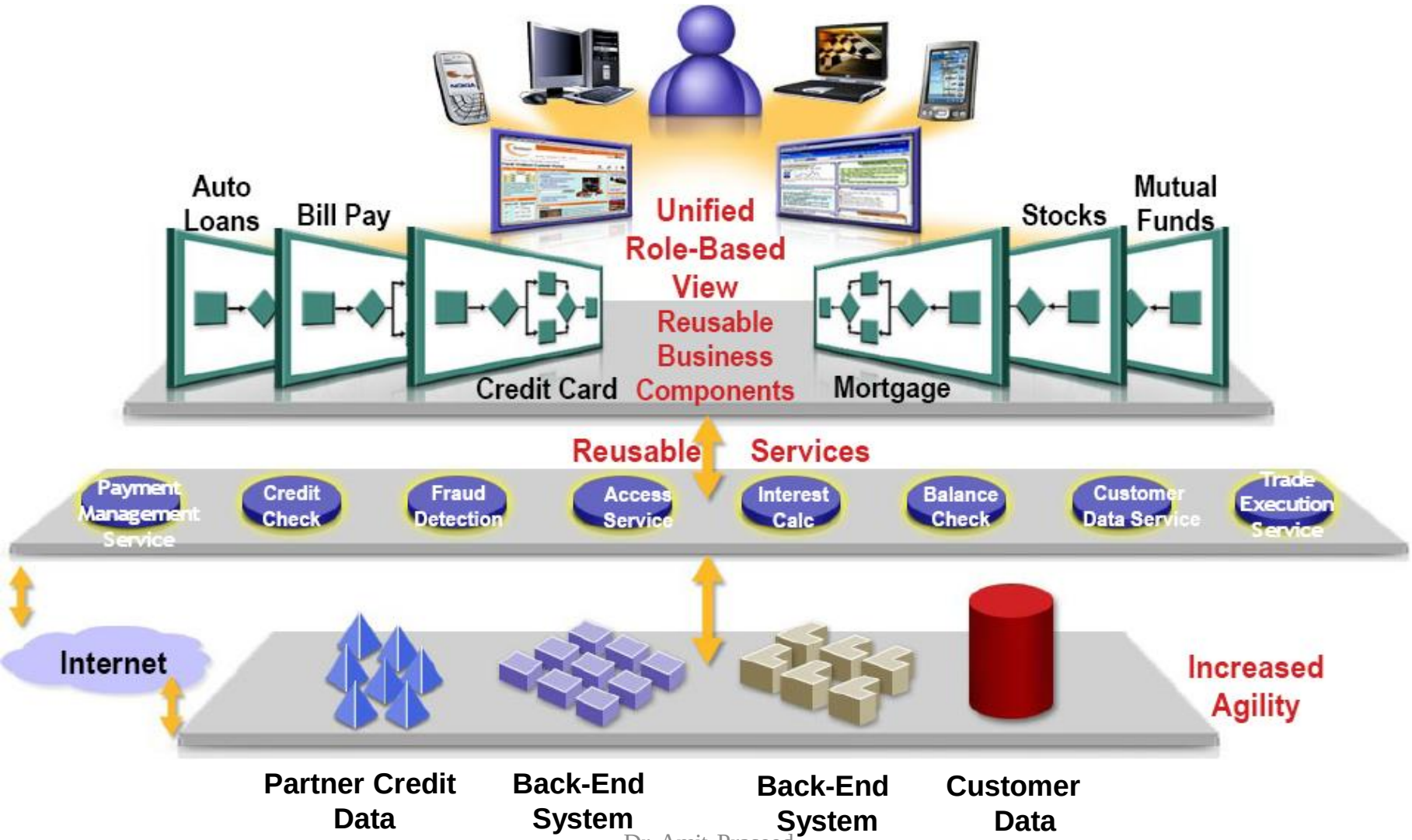
Partner Credit Data

Partner Credit Data

Backend System

Customer Data

A Better Architecture

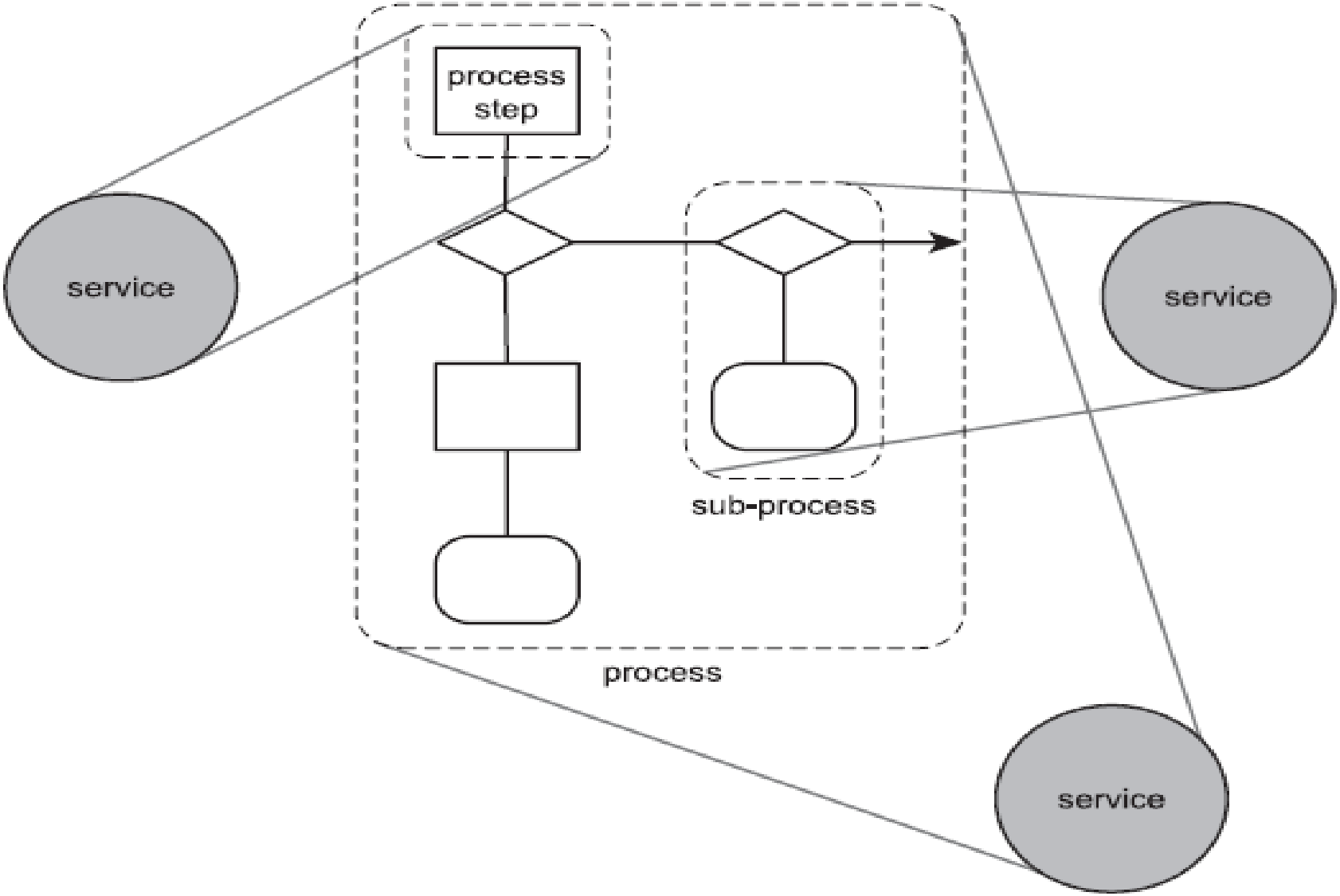


Service Oriented Architecture (SOA)

- Service-Oriented Architecture (SOA) is an architectural style.
- Applications built using an SOA style deliver functionality as **services that can be used or reused** when building applications or integrating within the enterprise or trading partners.

What constitutes a Service?

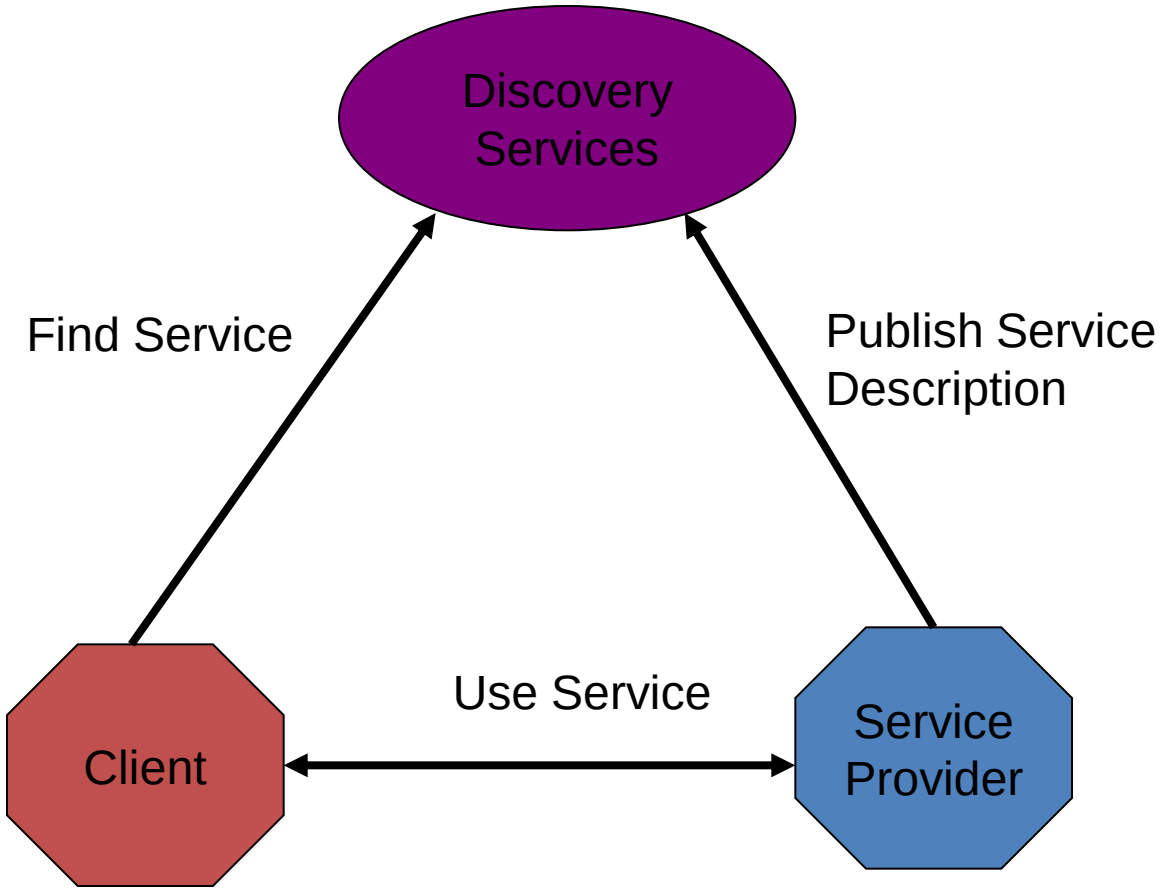
- A logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit, provide weather data, consolidate drilling reports).
- Self-contained.
- *May be* composed of other services
- Is a “black box” to consumers of the service.


process
step
service
service
sub-process
process
service

SOA Features

- *Loose coupling*
- *Service contract*
- *Autonomy*
- *Abstraction*
- *Reusability*
- *Statelessness*
- *Discoverability*

SOA Framework



Web Services

- A web service is “a software system designed to support interoperable machine-to-machine interaction over a network”
- It differs from a web application in one key point
  - Web applications enable communication between human clients and machines
  - Web services enable communication between clients

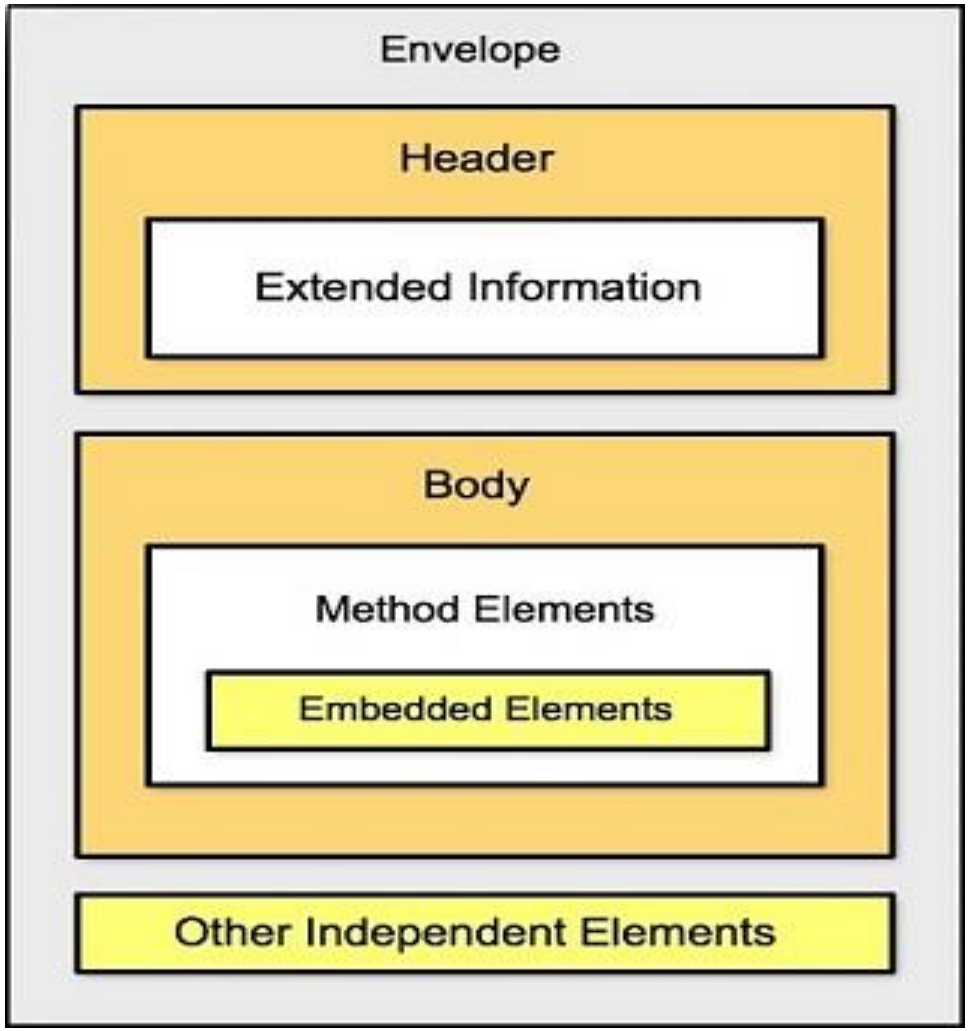
Composition of Web Services

- SOAP – Simple Object Access Protocol
  - XML based protocol used for communication
- WSDL - Web Services Description Language
  - describes the interface, a set of operations supported by a web service in a standard format.
- UDDI - Universal Description, Discovery, and Integration
  - provides a global registry for advertising and discovery of web services

SOAP Message Example

```
<?xml version="1.0"?>
<SOAP:Envelope
    xmlns:xsi="http://www.w3.org/1999/XMLSchema/instance"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema/instance"
    xmlns:SOAP="urn:schemas-xmlsoap-org:soap.v1">
  <SOAP:Body>
      <calculateArea>
        <origin>
           <x xsd:type="float">10</x>
           <y xsd:type="float">20</y>
        </origin>
        <corner>
           <x xsd:type="float">100</x>
           <y xsd:type="float">200</y>
        </corner>
       </calculateArea>
   </SOAP:Body>
</SOAP:Envelope>
```

SOAP Message Structure



SOAP Protocol Extensions

- WS-* is used to refer to extensions to the basic Web services framework established by first-generation standards
  - **WS-Security**: Handles encryption and digital signatures
  - **WS-Policy**: Expands on WS-Security, enabling you to more specifically detail how and by whom a service can be used.
  - **WS-I**: Provides a set of standards and practices to prevent interoperability issues, as well as standardized tests to check for problems.
  - **WS-BPEL**: Provides a way to specify interactions between multiple web services, such as branching and concurrent processing

Issues with SOAP

- SOAP was meant to facilitate easier communication between machines using a common platform of XML
  - Difficult to debug
  - Difficult to build
  - Difficult to use
- Salesforce had built an “Internet as a Service” package for selling APIs, but their use of complex XML proved to be a major deterrent.

REST

- A simpler way to implement SOA is using REpresentational State Transfer (REST)
  - Resource Identification through URI
  - Uniform, Constrained Interface
  - Self-Descriptive Message
  - Stateless Interactions
- The use of REST made API calls effortless, leading to companies like EBay, Amazon and Flickr cashing in

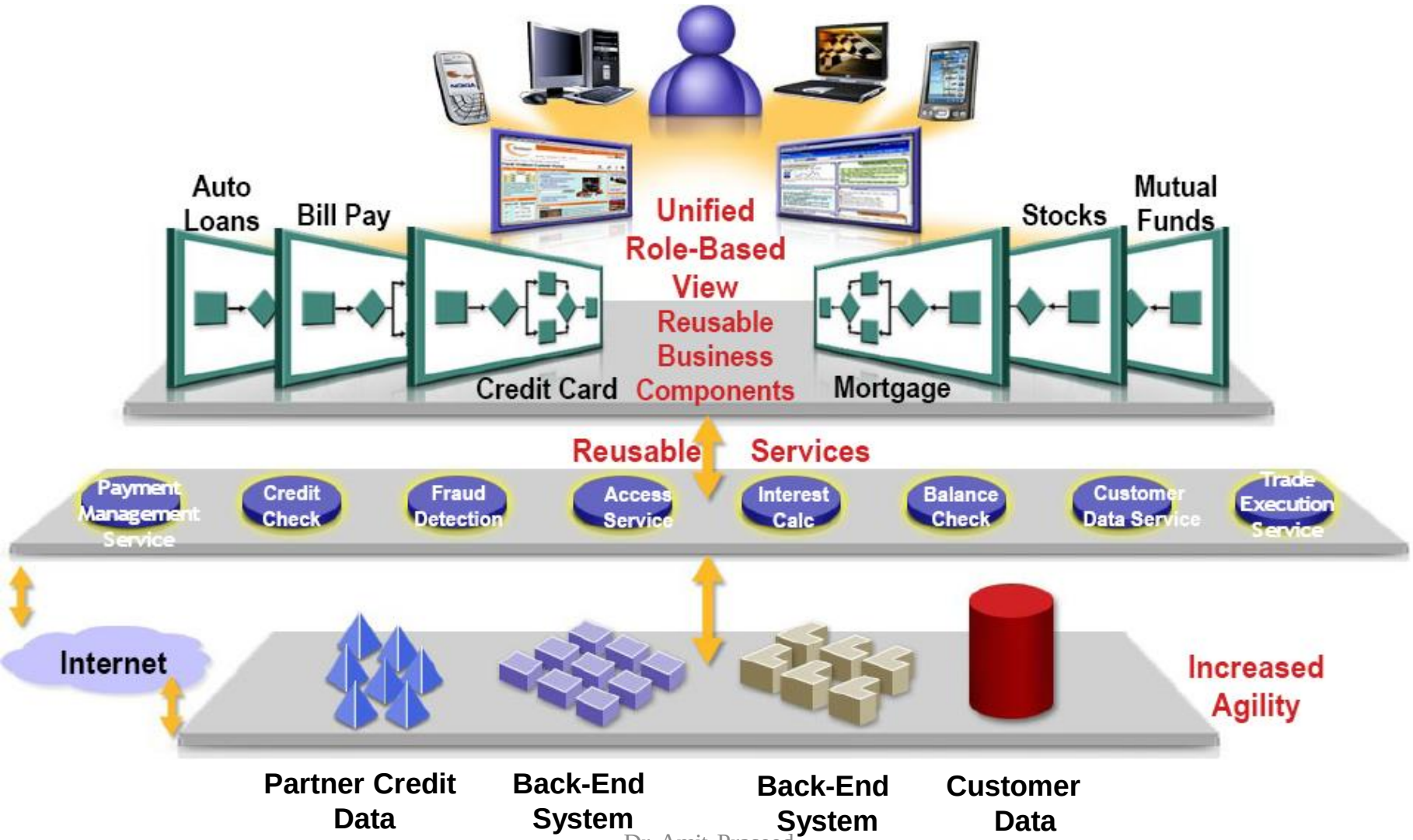
A Comparison

Feature	SOAP	REST
Nature	Protocol	Architectural Style
State	Stateful/Stateless	Stateless
Format	XML	XML, JSON, plaintext etc.
Transfer Protocol	HTTP, HTTPS, TCP, FTP, SMTP etc.	HTTP, HTTPS
Security	WS-Security, ACID, HTTPS, SSL	HTTPS, SSL
Speed	Slow	Fast
Learning Curve	Difficult	Easy
Community	Small	Large

So, is SOAP dead?

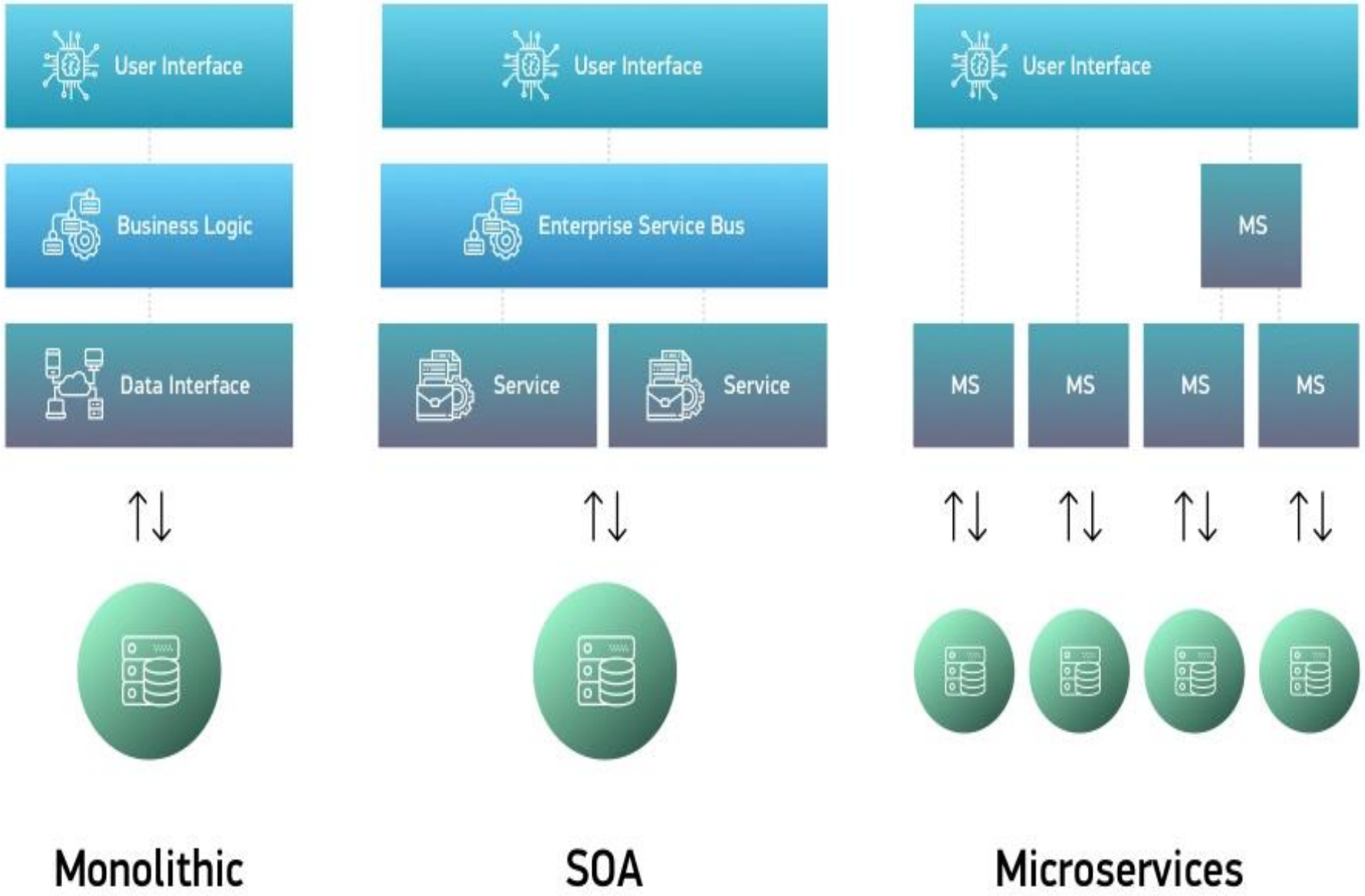
- REST is a simple and efficient way to build and use a Service Oriented Architecture
- Currently, nearly 83% of APIs use REST
- However, SOAP does have its benefits
  - Inbuilt successful/retry mechanism
  - Inbuilt security features
  - Extensible
  - Customizable

A Look Back at the SOA Architecture



Microservices : The Next Evolution

- Microservice Architecture involves the splitting of large software applications into decoupled modules, running unique processes and communicating through APIs
- Every microservice is an independent software component by itself, so they do not have to be modified frequently
- Almost all popular organizations like EBay, Amazon and Netflix employ microservices


User Interface
Business Logic
Data Interface
User Interface
Enterprise Service Bus
Service
Service
User Interface
MS
MS
MS
MS
MS
Monolithic
SOA
Microservices

Service Registration and Discovery

- Registration
  - Self Registration
  - Third Party Registration
- Discovery
  - Client Side Discovery
  - Server Side Discovery