ASU IRA A. FULTON SCHOOLS OF engineering

school of computing, informatics, & decision systems engineering

CSE 446 / CSE 598

Software Integration and Engineering

Unit 1

Service Standards and Service Development

Lecture 1-1

Introduction

Dr. Yinong Chen
https://myasucourses.asu.edu/

Roadmap of Unit 1

1. Introduction
   - **Software Integration**
   - **Service-Oriented Architecture**
   - **Service-Oriented Computing**
   - **Service-Oriented Development**
   - **Semantic Web and Intelligent Web**
2. Self-Hosting Services
3. Advanced Services
4. REST Concepts
5. RESTful Services
6. Advanced Web Application Architecture

What is a System?

A **system** is a collection of interconnected components that have some functions or purposes. A system can be defined at three levels

- A system has **architecture**
  - the overall organization of its components
- A system has **interface**
  - input/output
- A system has **behavior**
  - declarative relationship between input and output;
  - procedural relationship: transformation from input to output.

Software Integration at Different Levels

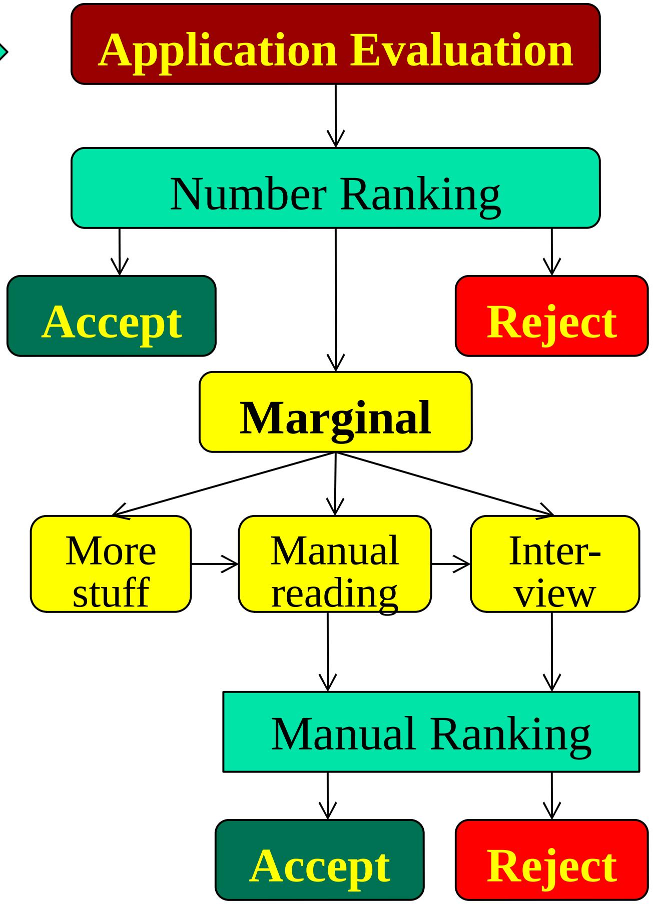
- **Architecture**
  - Integration of systems
  - Integration of components
  - Integration of computing systems and data sources

The distinction may or may not be clear, depending on the context.

- **Interface**
  - Integration based on component models
    - *Object Model (migration or remote objects)*
    - *Service Model*
  - Integration based on communication model
    - *Point to point (Remote procedure call and invocation)*
    - *Bus (object broker and service bus)*
- **Behavior**
  - Integration based on declarative relationship among inputs and outputs: Integration through message exchange among components (services and objects).
  - Integration based on workflow among the components

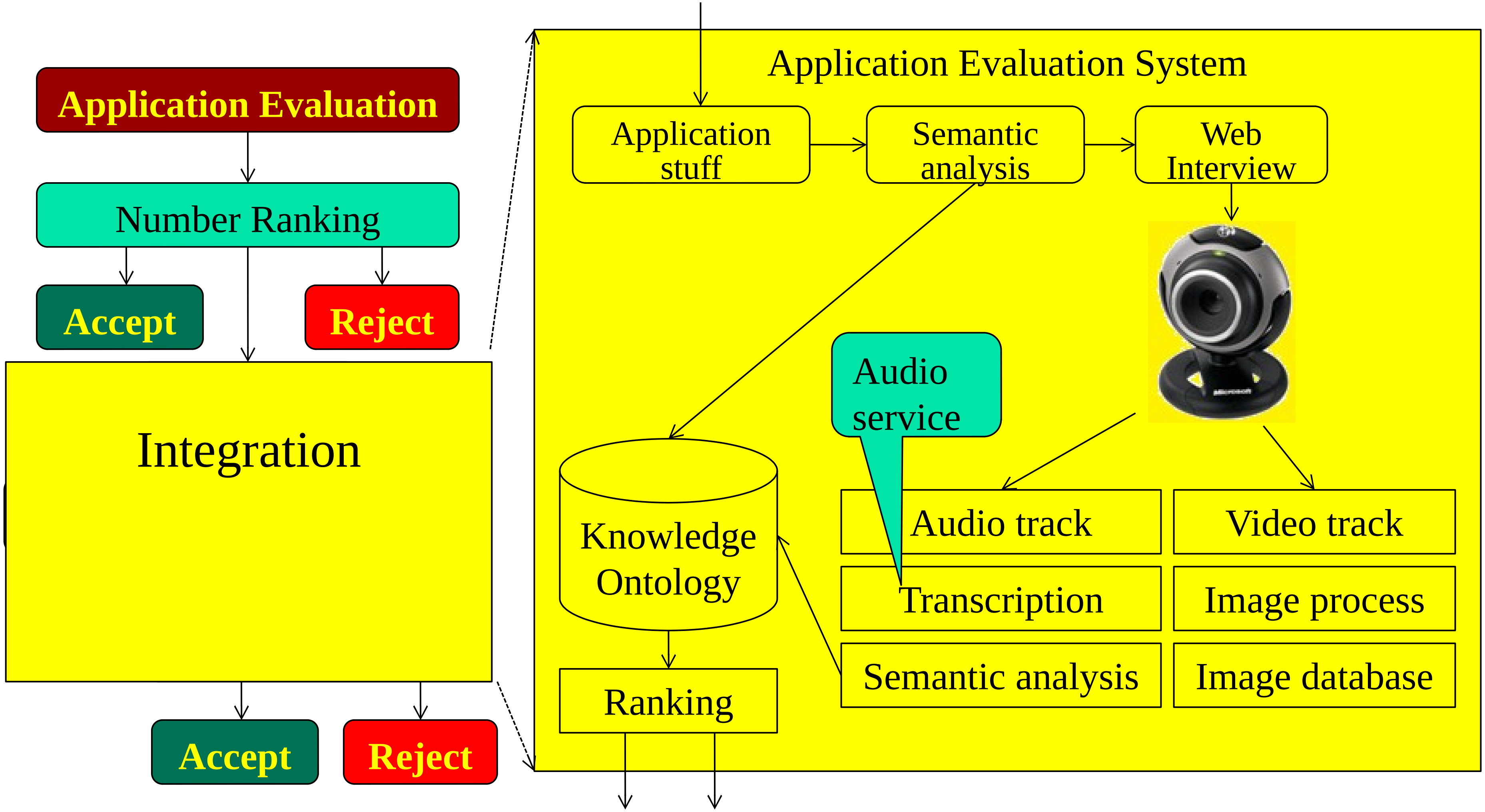
Integration Case Study: ASU Website

- Enrollment
  - Application submission
  - Application evaluation
- My ASU: Canvas
  - Class search and enrollment
  - Class schedule
  - Test and assignment submission
  - Grade book
  - Mail and alert
- Student database
- ASU Gmail
  - Mail
  - Calendar and schedule



Integration Case Study: ASU Website

What else can we do to improve our business?



Application Evaluation

Number Ranking

Accept

Reject

Integration

Accept

Reject

Application Evaluation System

Application stuff

Semantic analysis

Web Interview

Audio service

Knowledge Ontology

Ranking

Audio track

Video track

Transcription

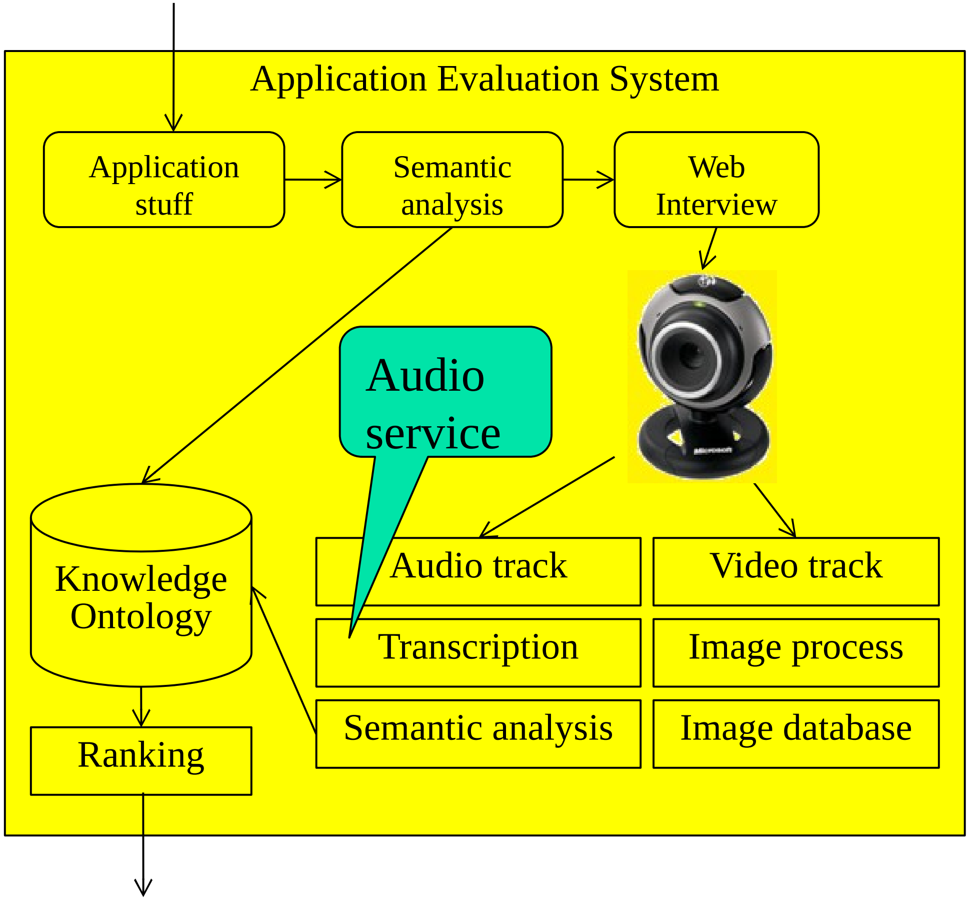
Image process

Semantic analysis

Image database

Starting Your Own Business

What you can do once you have developed the system?



- Start your own HR business to help any company to do prescreening;
- Integrate your system into any organization’s IT system;
- Make each component available as services to integrate into other system, e.g., your ontology, transcription service, or image processing;
- Deploy your system into a cloud environment, so that you can focus on improving your business logic, instead of working on hosting business.
- What is the Amazon model?

Definitions and Terminologies

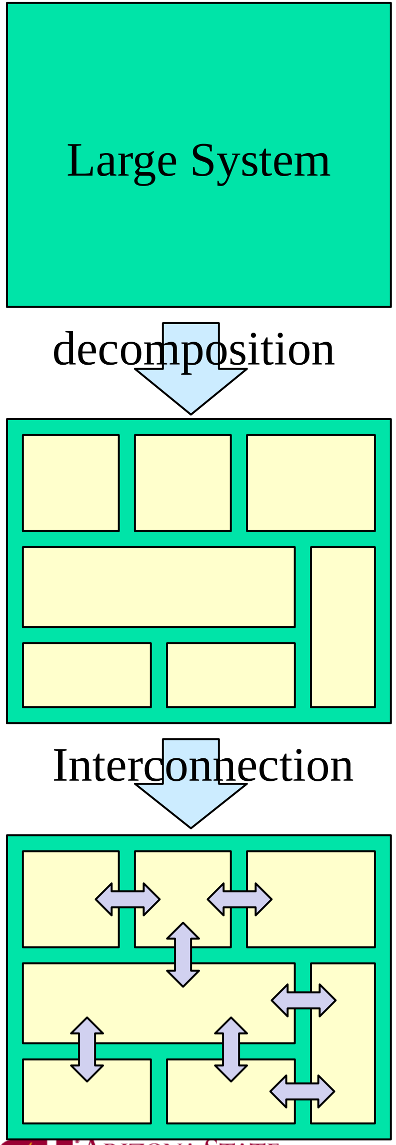
- **Service Orientation** (SO): Umbrella concept for all service-oriented approaches, including SOA, SOC, and SOD.
- **Service-Oriented Architecture** (SOA): Software consisting of a collection of loosely coupled and platform-independent services that interact with each other through standard interfaces. SOA does not concern developing operational software.
- **Service-Oriented Computing** (SOC) refers to the paradigm that represents computation in SOA. SOC concerns interface and behavior, including communication protocols, algorithms, and data representations.

SOA and SOC can be used alternatively if the entire system is concerned, without specifically referring to architecture, interface, or behavior.

Definitions and Terminologies (contd.)

- **Service-Oriented Development** (SOD)
  - concerns the entire software **development life cycle** based on SOA concepts and SOC paradigm.
  - involves current technologies and tools to effectively produce operational software:
    - Programming language features,
    - Development environments: VS, Java EE, WebSphere;
    - Standards and Interfaces with components, services, and data sources;
    - Deployment and hosting environment: Self-Hosting, Server, Cloud Infrastructure, such as Google App Engine, MS Azure, etc.

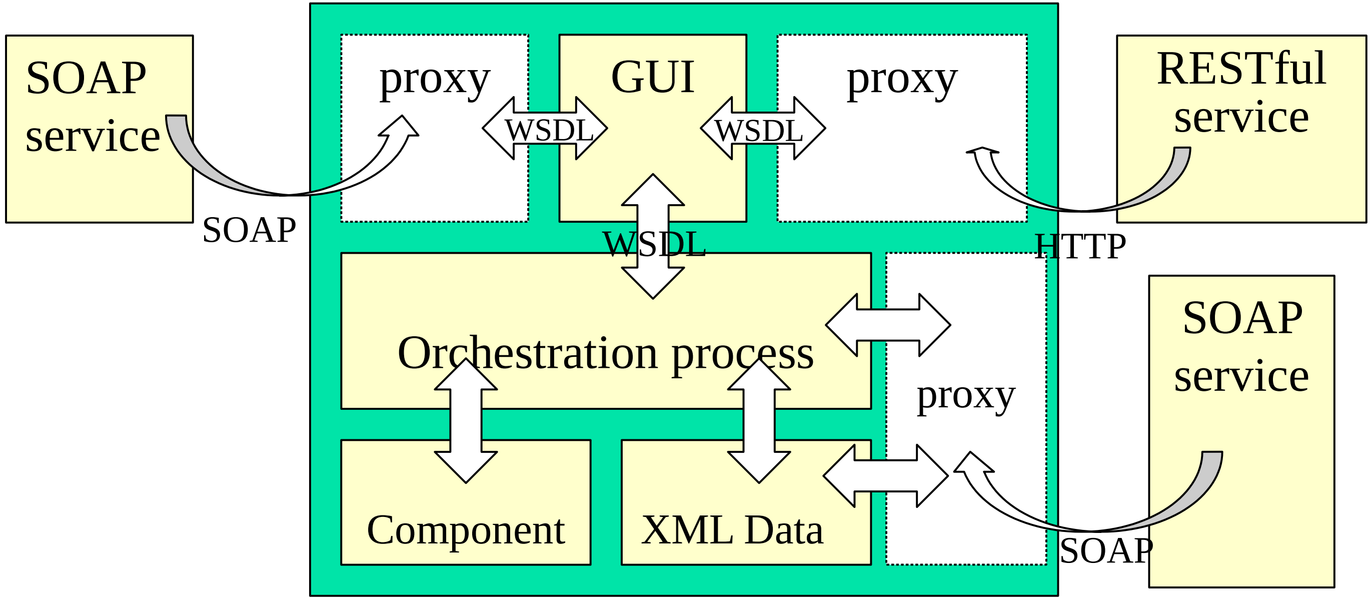
SOA SOC (adding behavior & Interface)



Large System

decomposition

Interconnection



SOAP service

SOAP

proxy

WSDL

GUI

WSDL

proxy

RESTful service

HTTP

WSDL

Orchestration process

proxy

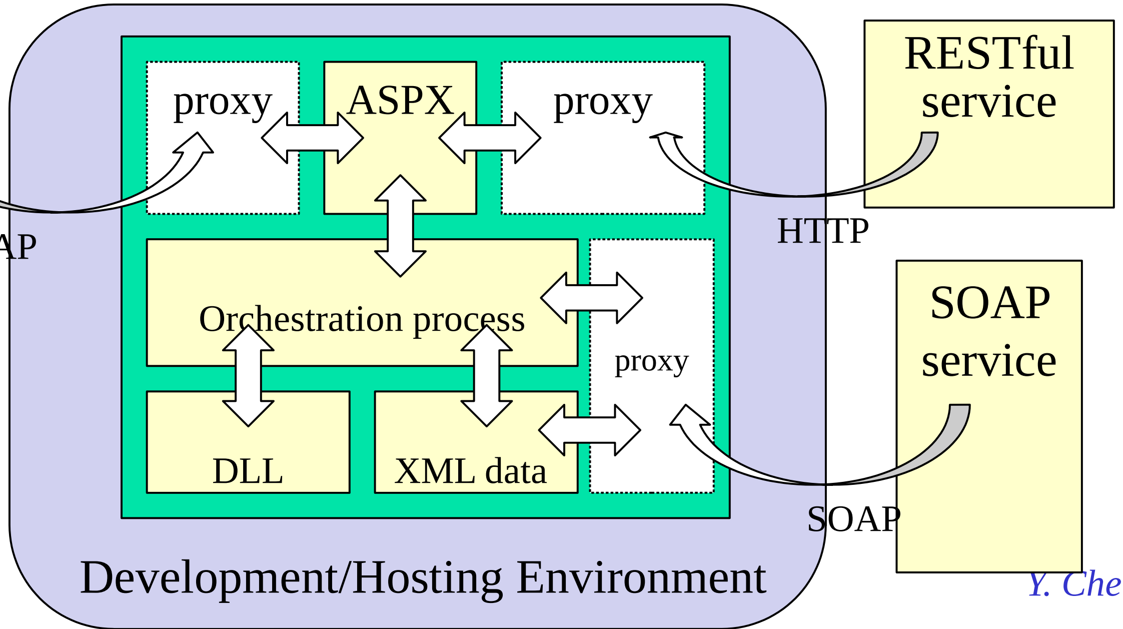
Component

XML Data

SOAP service

SOAP

SOD



SOAP service

SOAP

proxy

ASPX

proxy

RESTful service

HTTP

Orchestration process

proxy

DLL

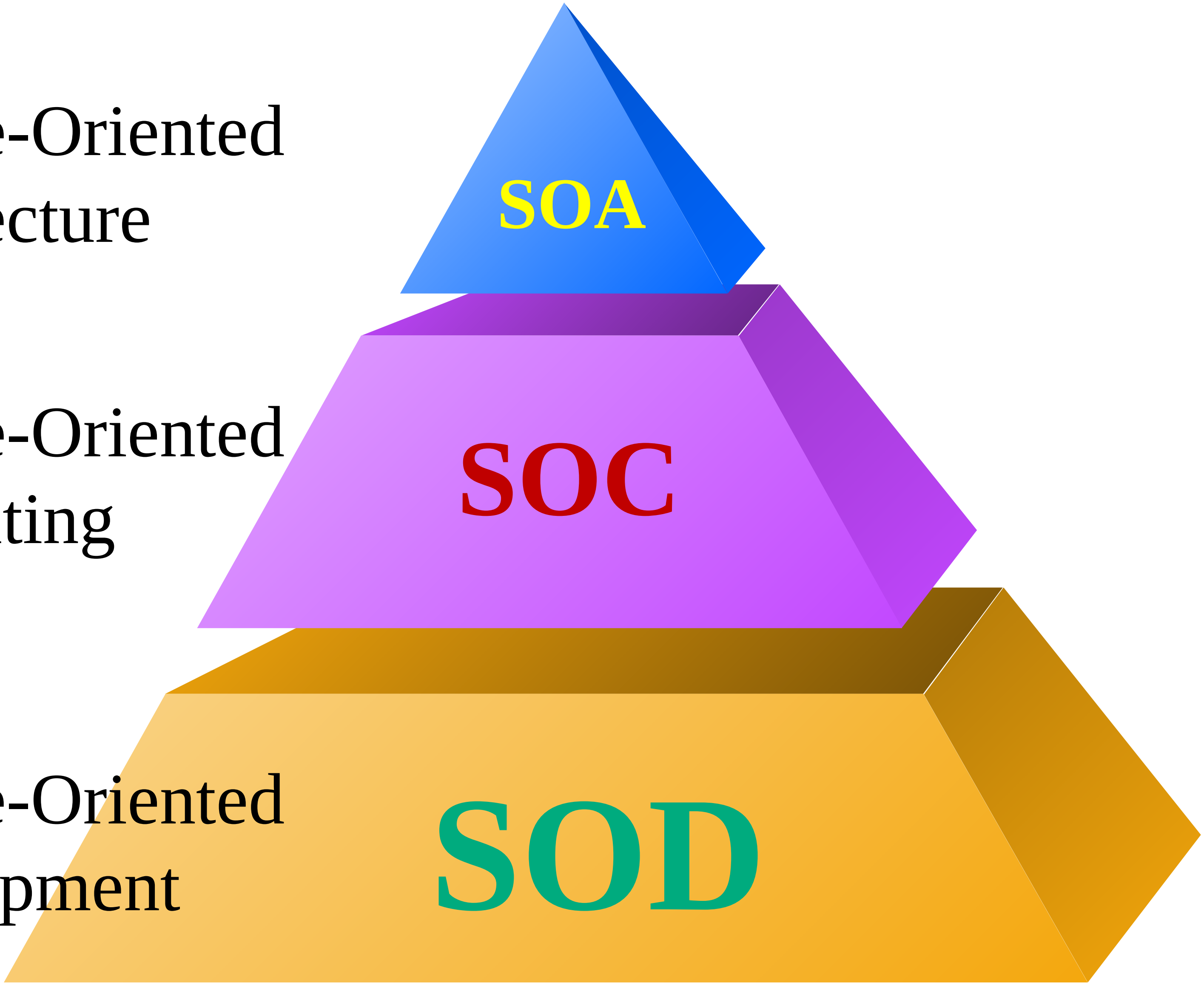
XML data

SOAP service

SOAP

Development/Hosting Environment

Service Orientation (SO)



Service-Oriented Architecture

SOA

Service-Oriented Computing

SOC

Service-Oriented Development

SOD

Why Service Orientation?

- Service orientation is similar to object orientation
- There is little difference at architecture level
- Driven by Development Requirements (developing operational software):
  - Large scalable and evolvable enterprise systems
  - Integration among existing systems
  - Interoperable among languages and platforms
  - Manageable complexity
  - Reusable business functions
  - Internet (Web) provisioning of services
  - Reduced cost of doing business
  - Increased dependability (reliability, availability, and security)

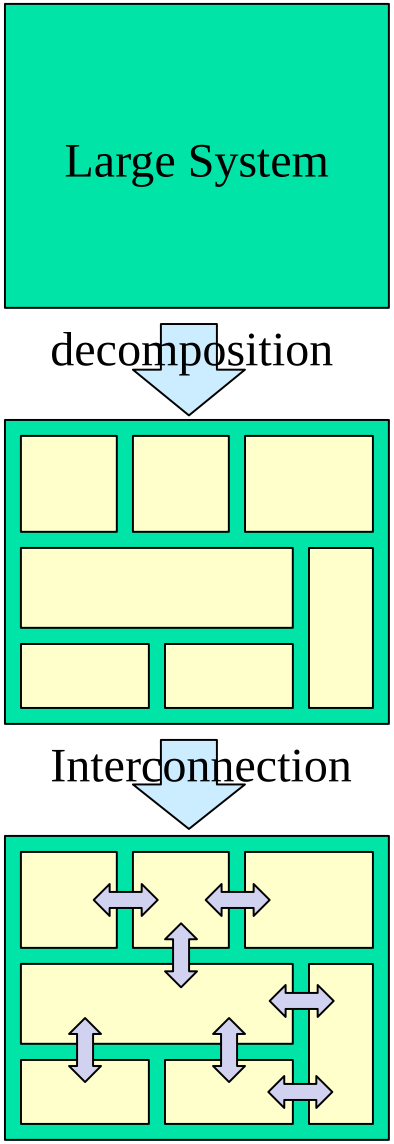
Why is SO Different from Object Orientation? Interface and Environment!

- OO has evolved into distributed object systems
  - CORBA, COM, DCOM, RMI (remote method invocation), etc. all try to implement *transparent* distributed systems
  - Standard interface such as IDL in CORBA
- SO standards are jointly developed by major players
  - IBM, Microsoft, Oracle, SAP, Sun Microsystems
  - SOAP, WSDL, HTTP, UDDI, URI
- Loosely coupling instead of tightly coupling
  - Message exchange instead of remote calls
  - Using proxy instead of code integration or migration
- Infrastructure for Service Publication (Service Brokers)
  - OO does not have standard publication mechanisms
  - SO enable and encourage sharing through standard brokers

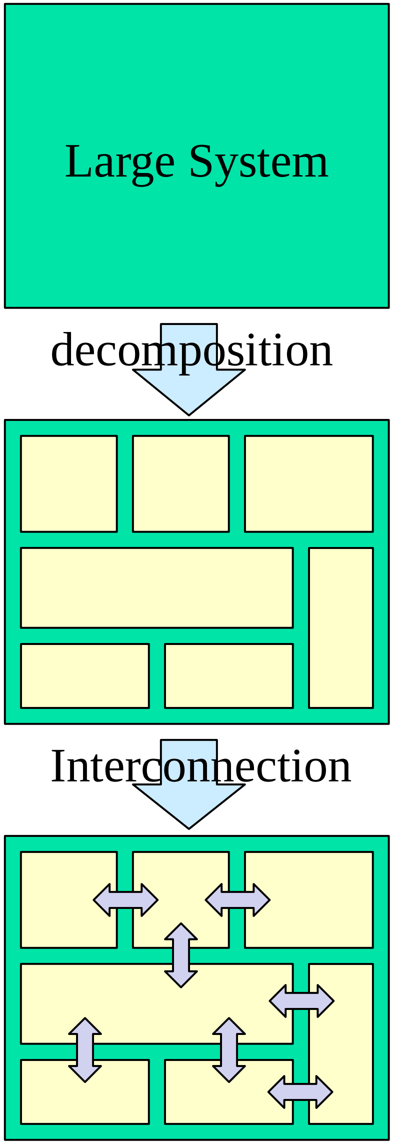
Better for integration

SOA vs. Object-Oriented Architecture (OOA)

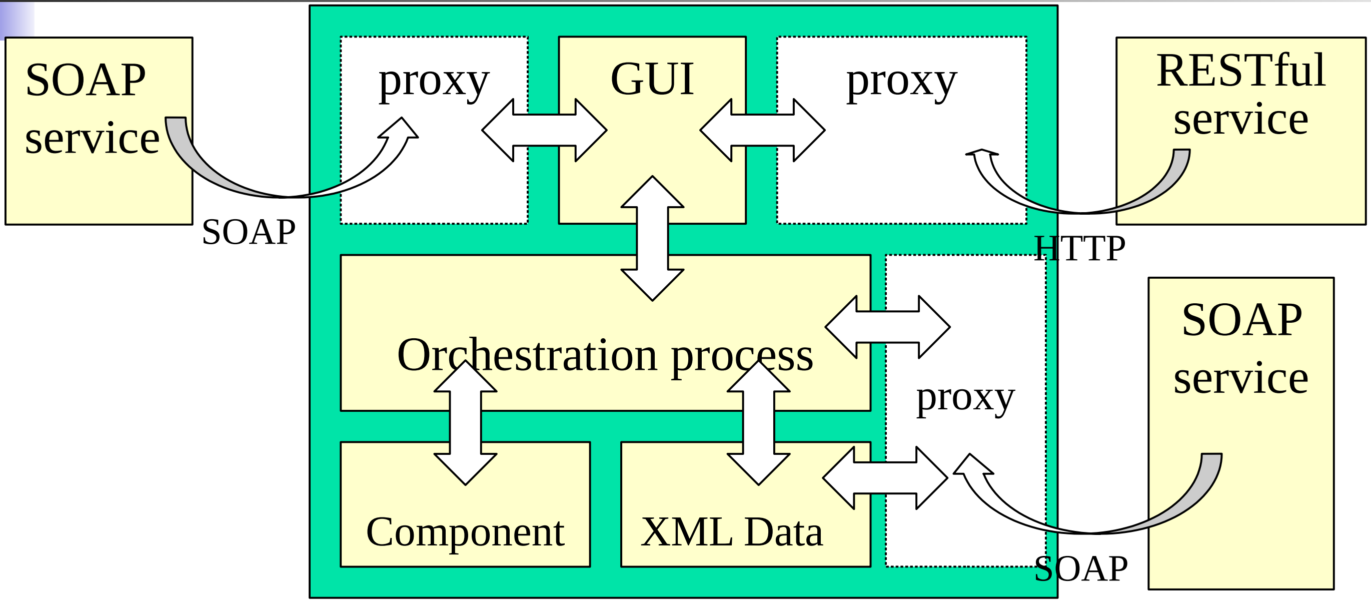
SOA

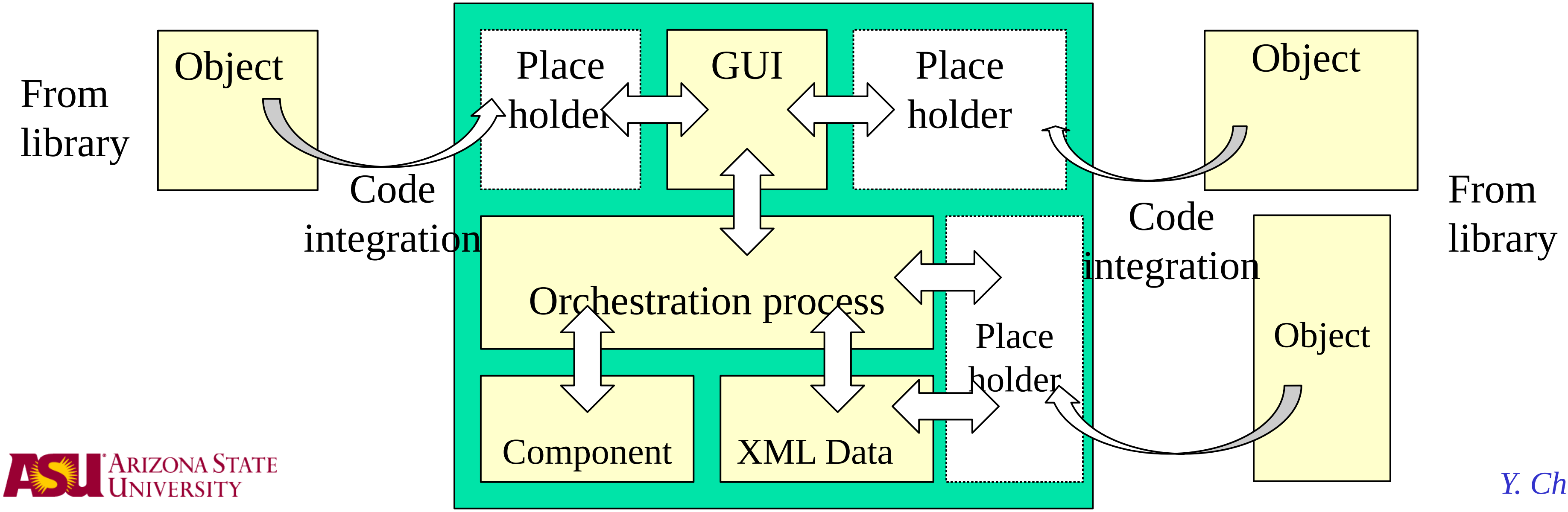


OOA

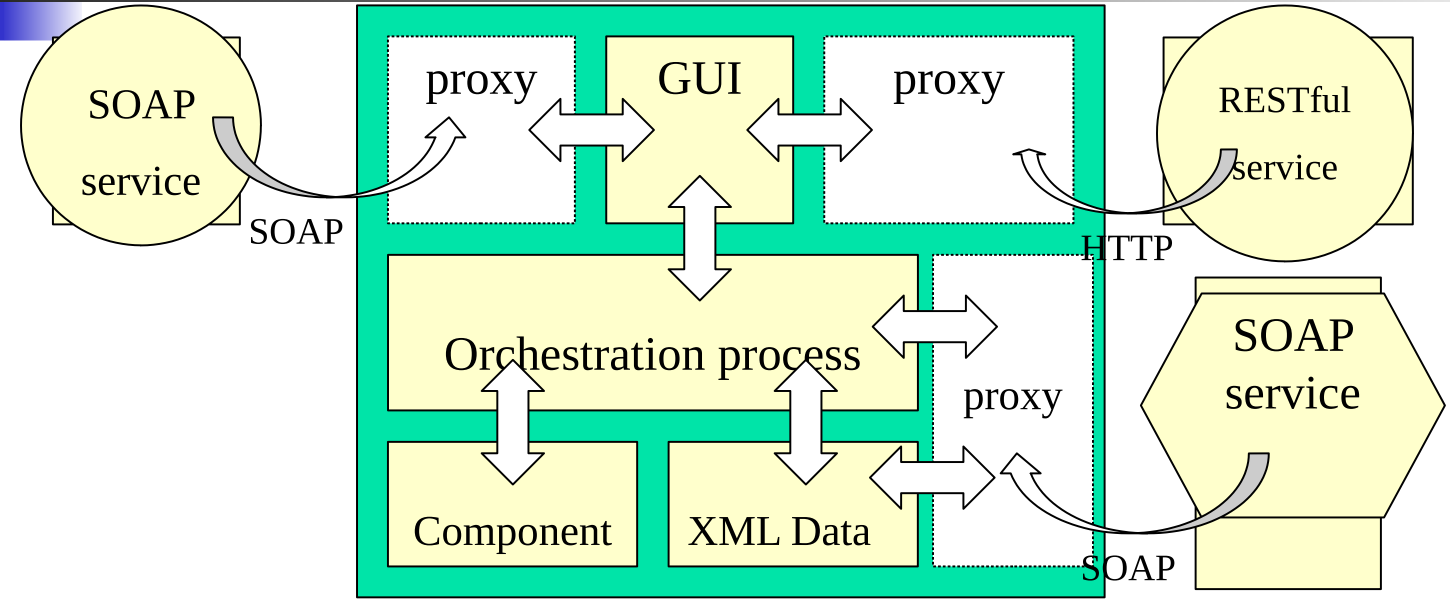


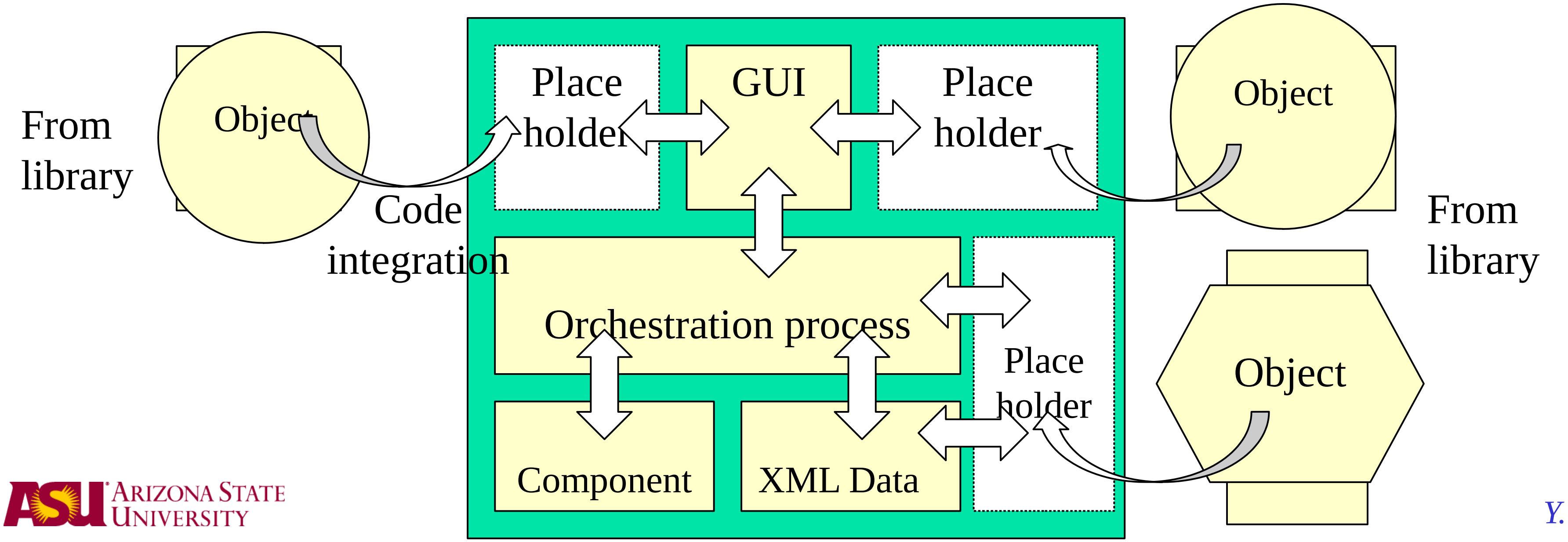
Interface & Behavior Levels: SOC vs. OOC





Interface & Behavior Levels: SOC vs. OOC





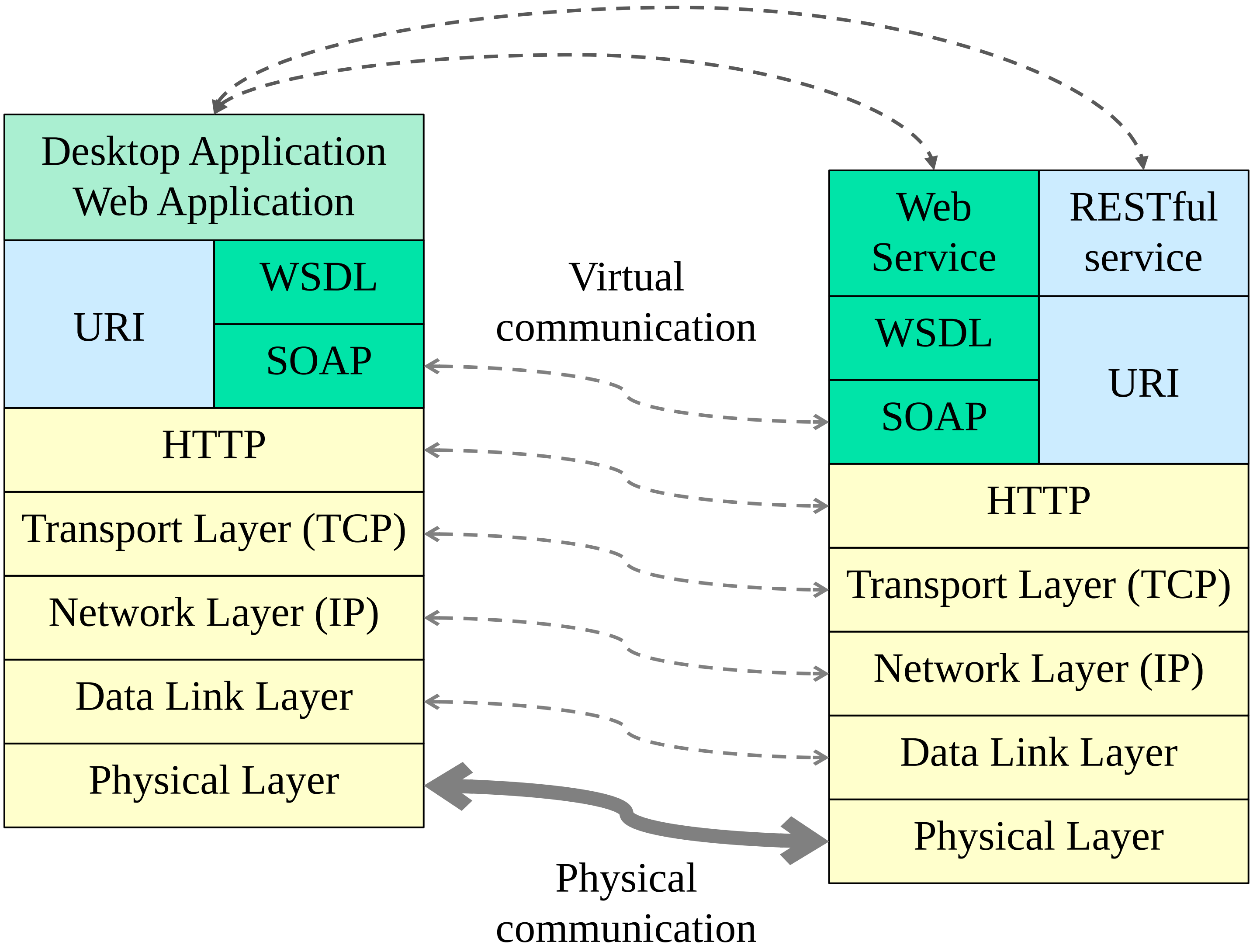
Definition of Web Service and SOAP Service

- A Web service is an **interface** that describes a **collection of operations** (Web methods) that are Web-accessible through standardized XML messaging
  - WSDL (Web Service Description Language) is used to describe the interface;
  - SOAP (Simple Object Access Protocol) is used to exchange messages between services and between the applications and services;
  - SOAP is typically on HTTP (others are possible)
  - UDDI and ebXML are used for service registration/publishing
- Web services use XML for representing input and data, as well as almost everything: SOAP, WSDL, UDDI, and ebXML are all in XML.
- Web services that use SOAP and WSDL are called “SOAP Services”.

Definition of RESTful Service

- REST: Representational State Transfer
- A RESTful service is an **interface** that describes a **collection of resources** that are Web-accessible through standardized resource representations
  - URI (Uniformed Resource Identifier) is used to access the resource under the Web;
  - HTTP (Hypertext Transport Protocol) is used to exchange messages between services and between the applications and services;
  - RESTful services use one layer of protocol less (without SOAP), and HTTP is even more widely used.
- RESTful services use multiple resource representation languages for representing input and output data:
  - XML, HTML, XHTML, RSS Feed, Atom Feed, JSON
  - Does not deal with object types, and less related to OO

Service Protocol Stack


Desktop Application
Web Application
URI
WSDL
SOAP
HTTP
Transport Layer (TCP)
Network Layer (IP)
Data Link Layer
Physical Layer
Virtual
communication
Web
Service
RESTful
service
WSDL
SOAP
URI
HTTP
Transport Layer (TCP)
Network Layer (IP)
Data Link Layer
Physical Layer
Physical
communication