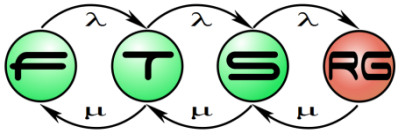
Webszolgáltatások implementációja

Ráth István

rath@mit.bme.hu

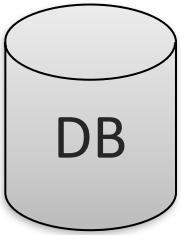


Házi feladat áttekintés

Házi feladat áttekintés

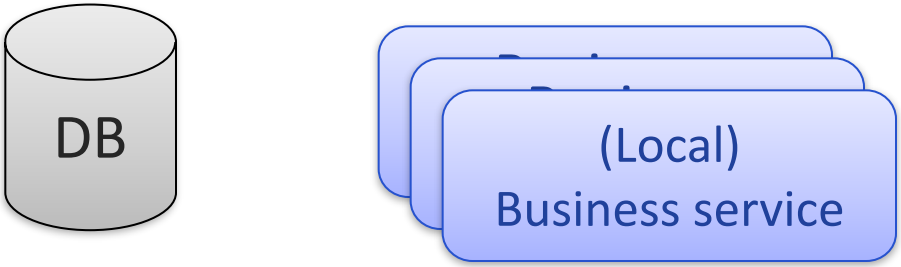
Workflow

Házi feladat áttekintés



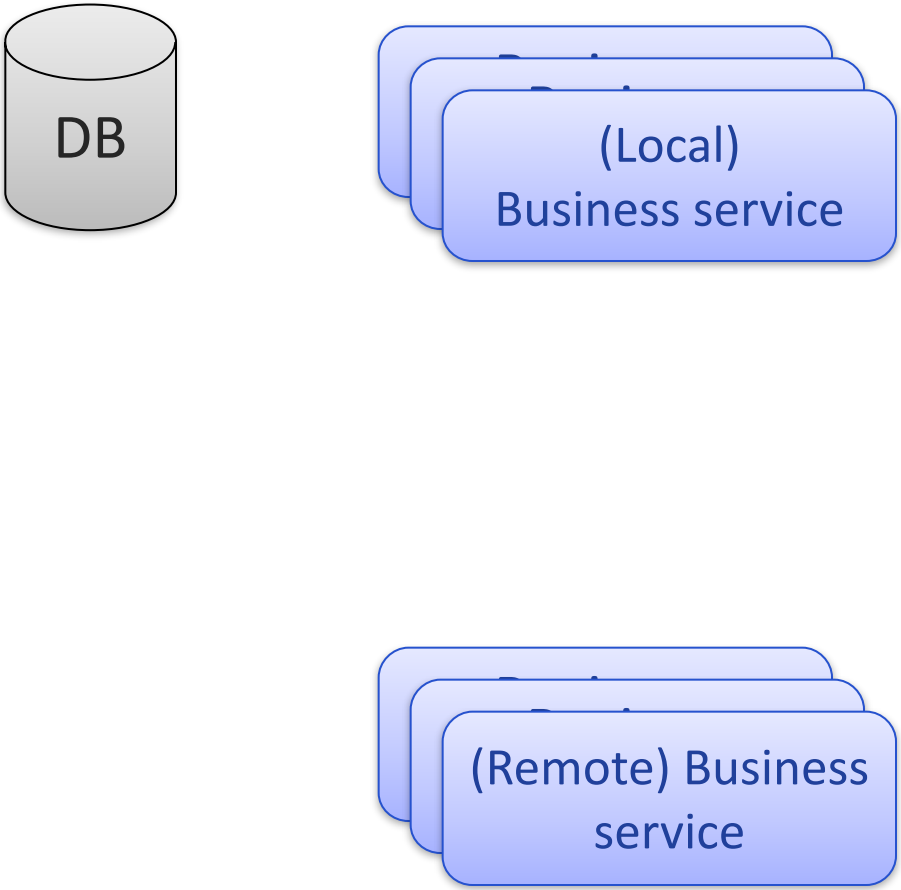
Workflow

Házi feladat áttekintés

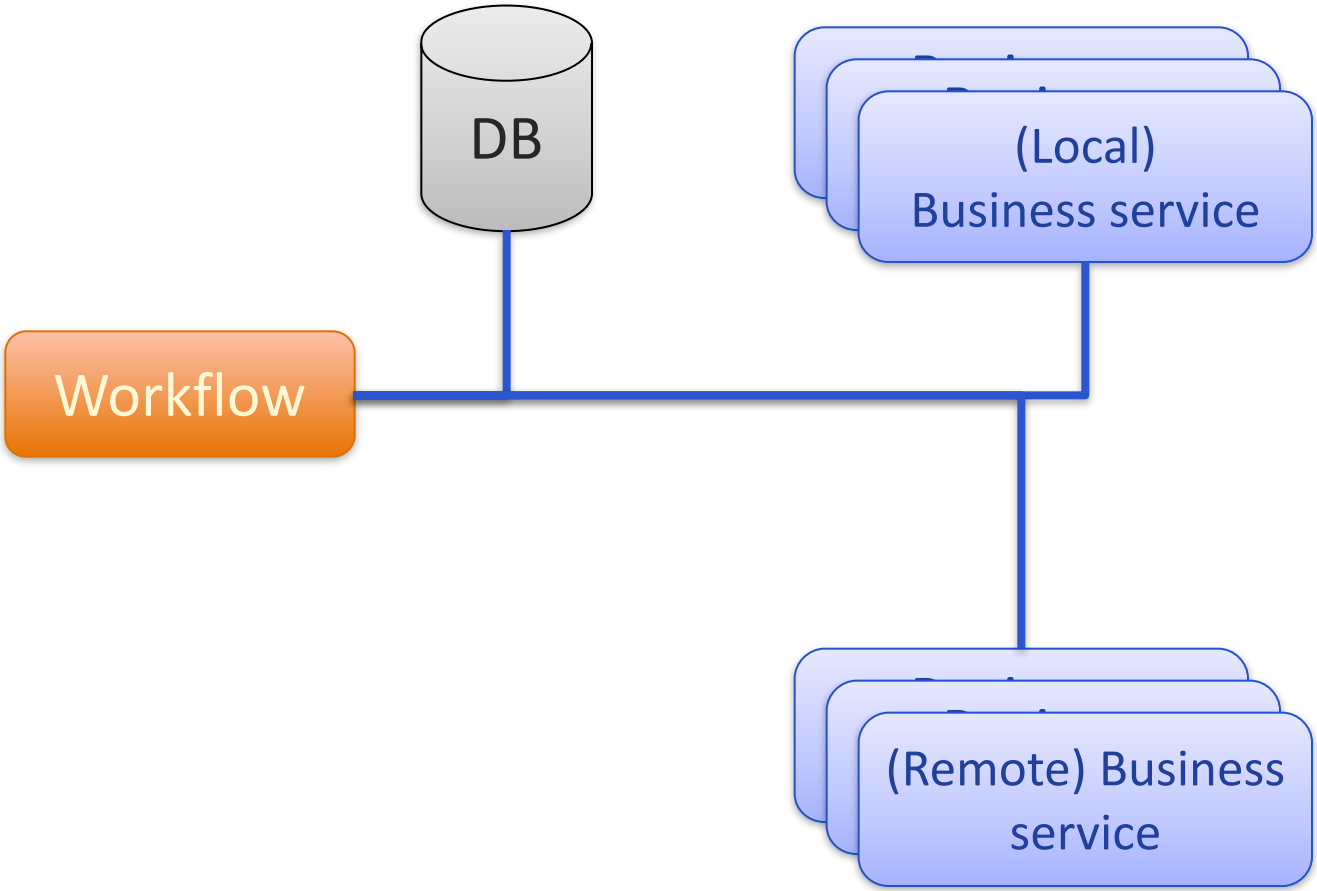


Workflow

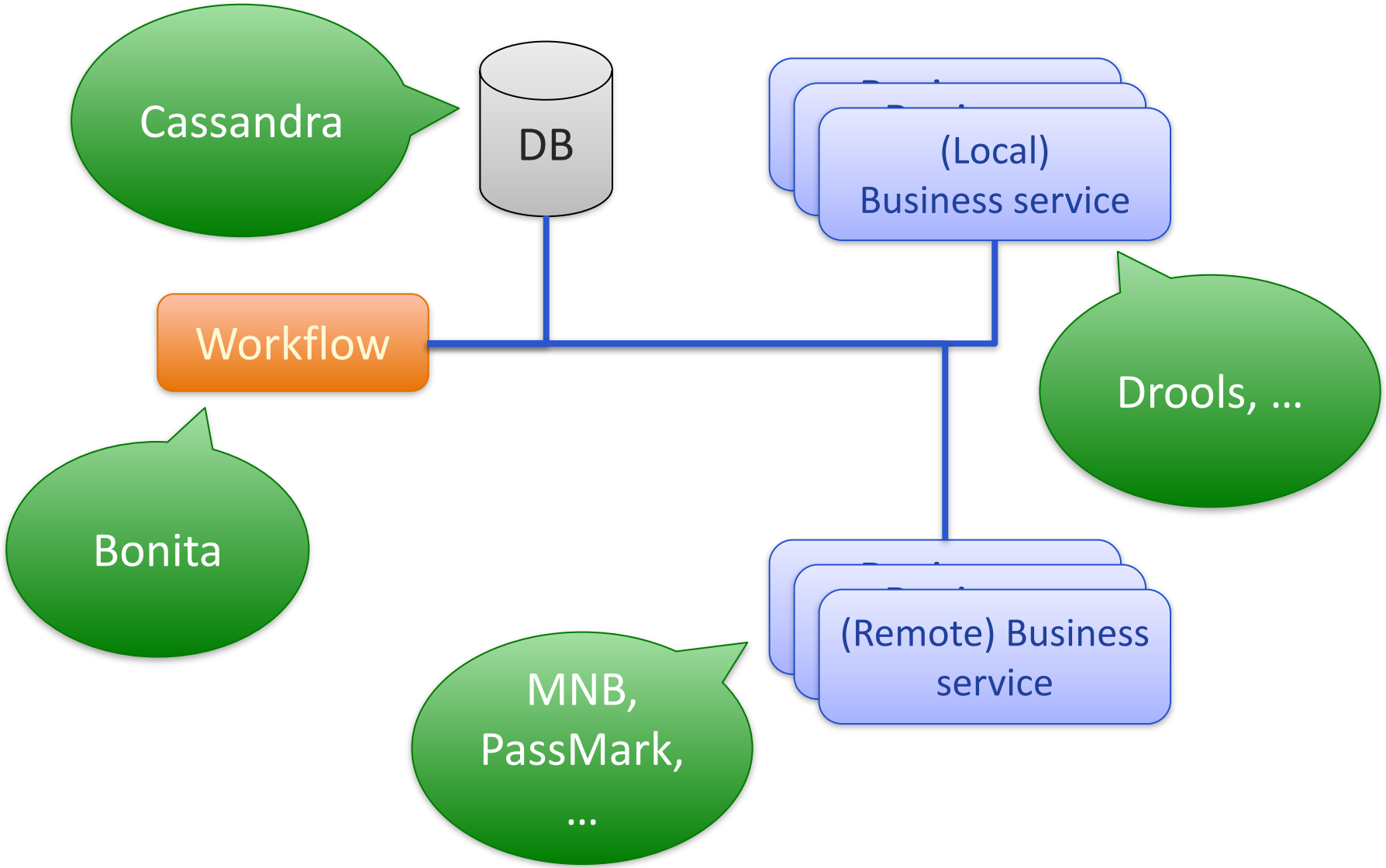
Házi feladat áttekintés



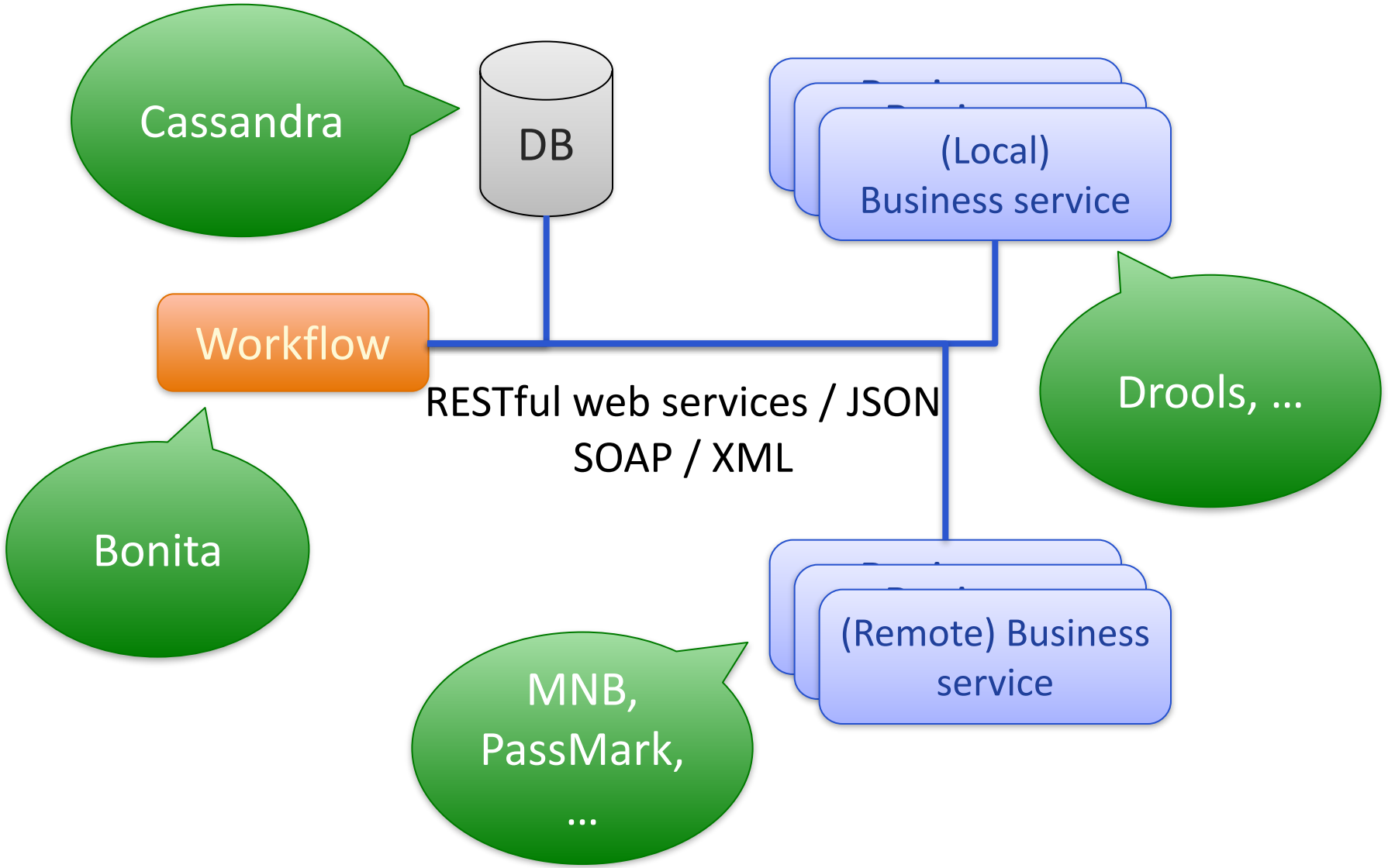
Házi feladat áttekintés



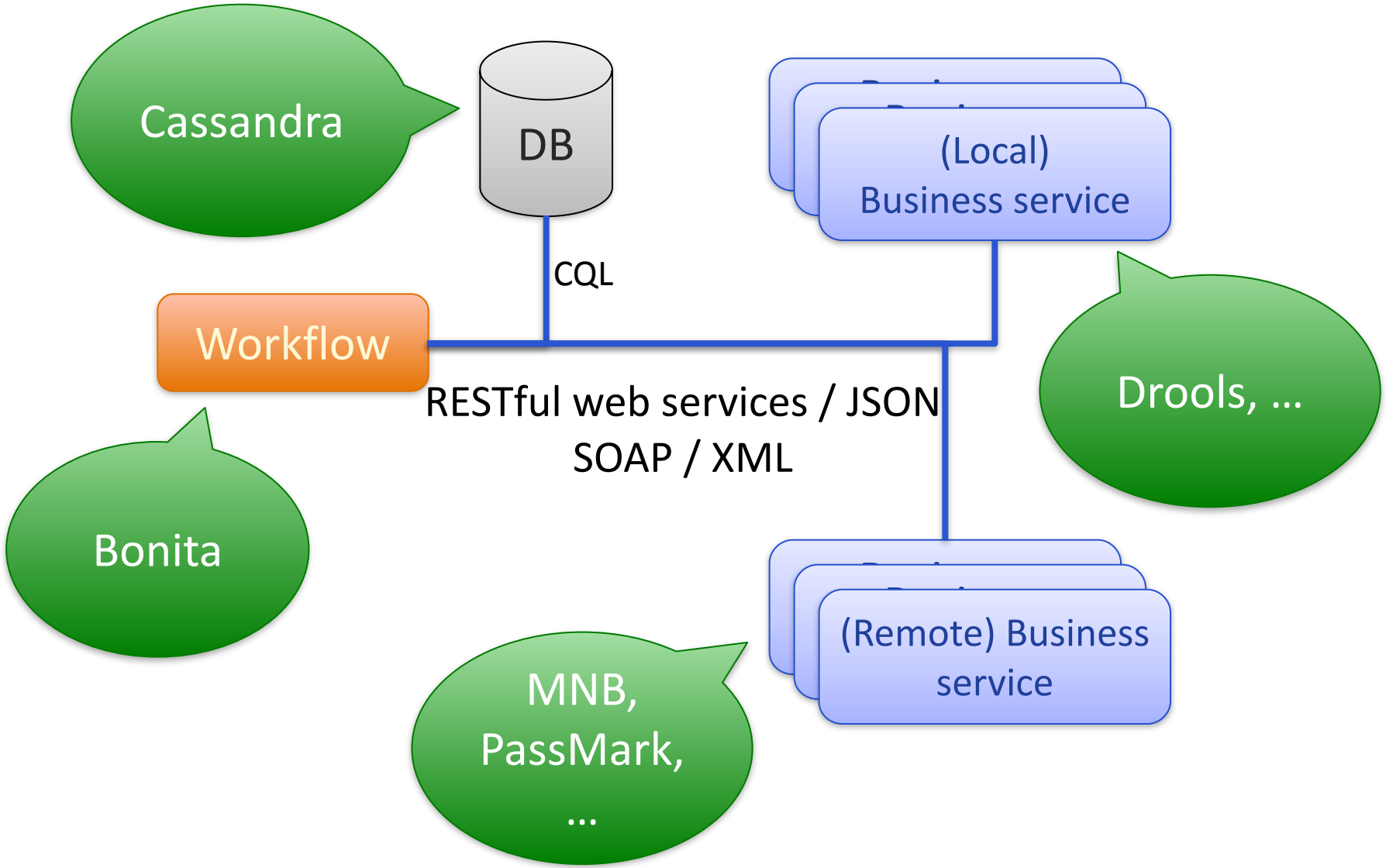
Házi feladat áttekintés



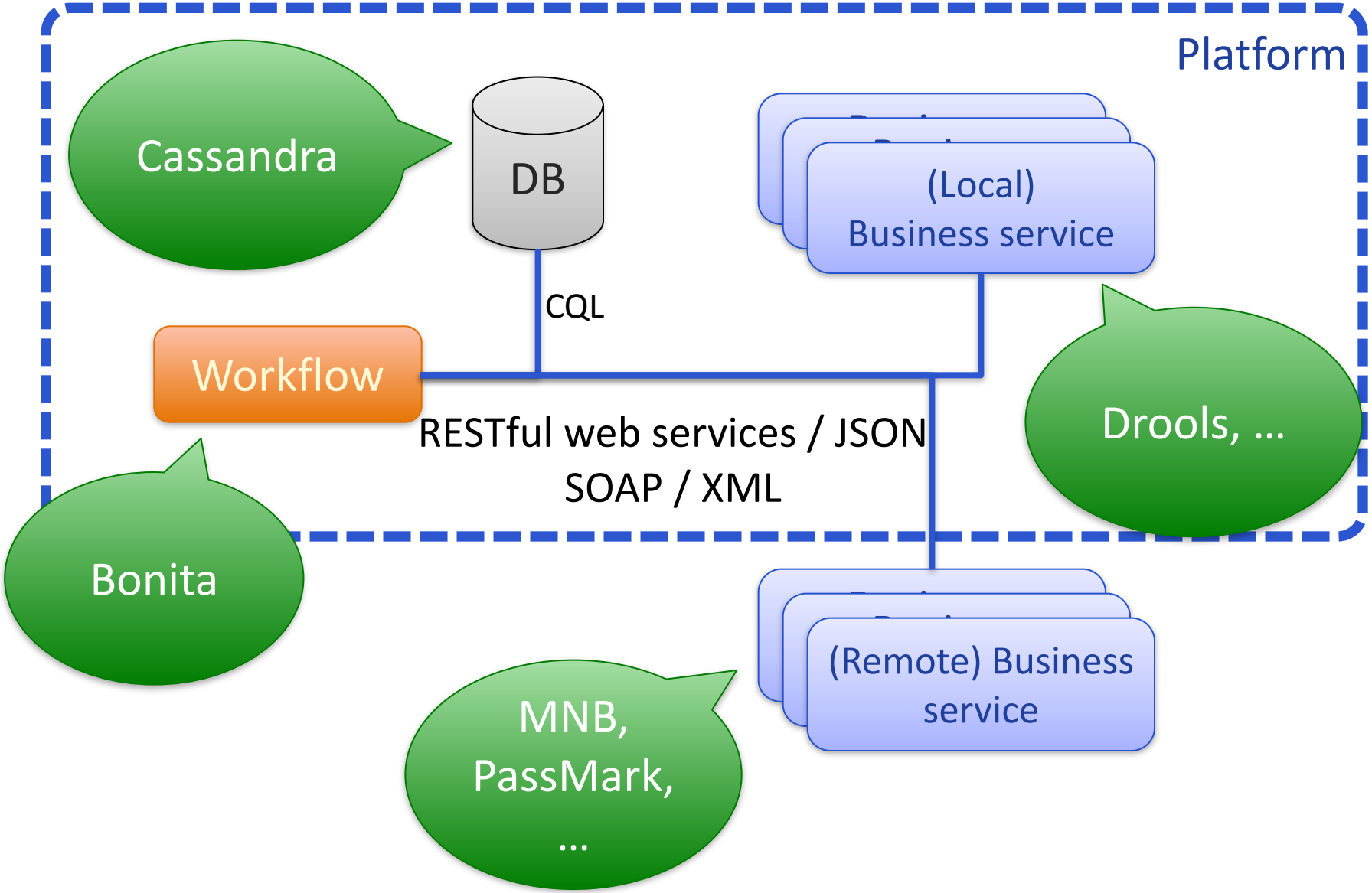
Házi feladat áttekintés



Házi feladat áttekintés



Házi feladat áttekintés

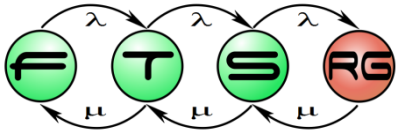


Tartalom

- Alapelvek
  - JSON és REST
- Konkrét technológiák
  - Java RESTful web services: JAXRS
  - Java SOAP web services: JAXWS
  - Java XML Bindings: JAXB
- Platform
  - Privát cloud: Eclipse Gyrex

JSON and REST

The New Kids on the Data Block



REST

What does it stand for?:

REST

What does it stand for?:

Representational **S**tat**e** **T**ransfer

REST

What does it stand for?:

Representational **S**tat**e** **T**ransfer

What Is it?

REST

What does it stand for?:

Representational **S**tat**e** **T**ransfer

What Is it?

A **style** of software architecture for distributed systems

REST

What does it stand for?:

Representational **S**tat**e** **T**ransfer

What Is it?

A **style** of software architecture for distributed systems

Who/Where/When?

REST

What does it stand for?:

Representational **S**tat**e** **T**ransfer

What Is it?

A **style** of software architecture for distributed systems

Who/Where/When?

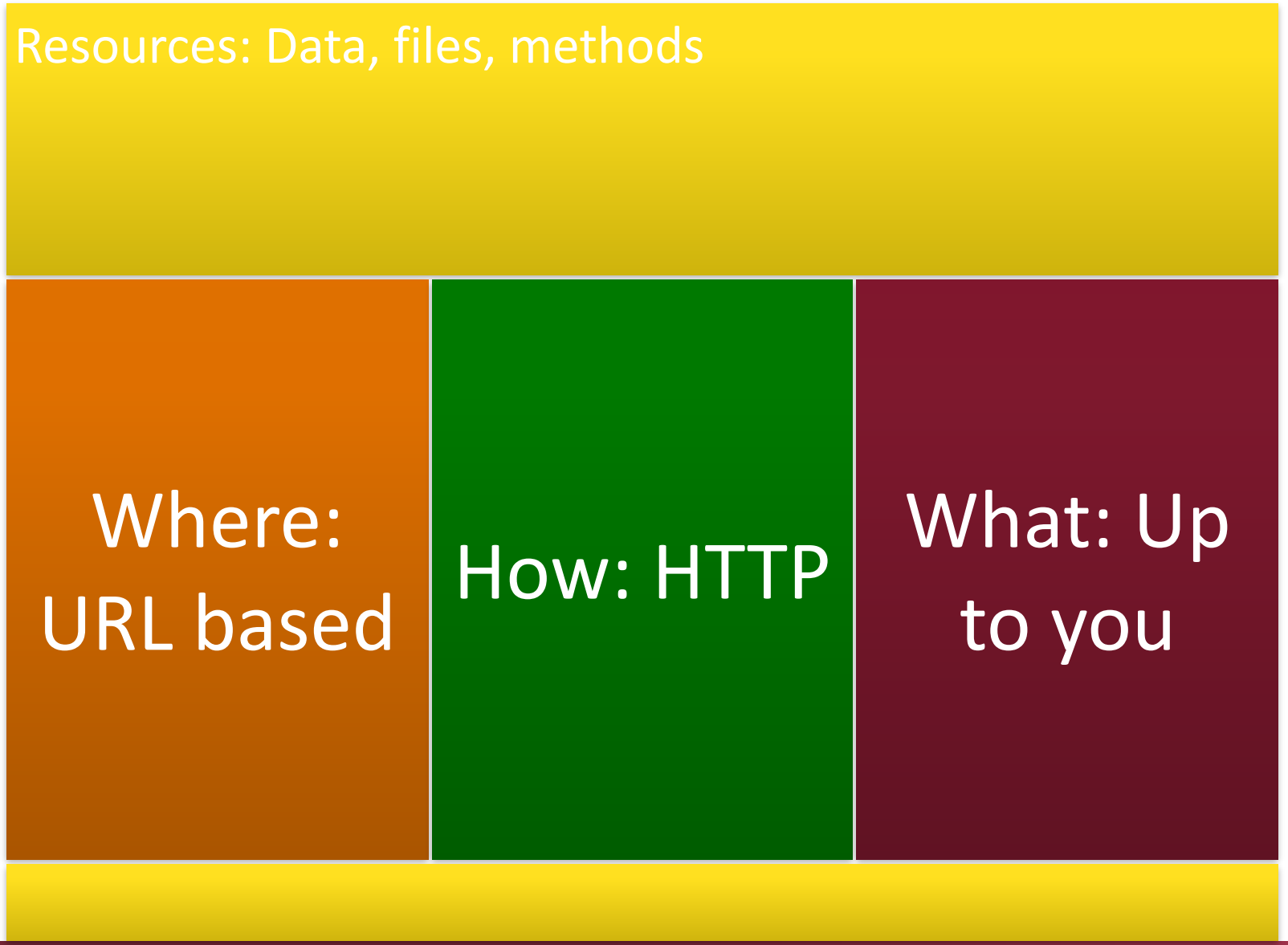
Came about in 2000 doctoral dissertation of Roy Fielding – but it's been used for much longer

Core principles

- Client-server
- Stateless
- Cacheable
- Layered
- Code on demand (optional)
- Uniform interfaces
  - URIs
  - Metadata
  - Self-descriptive

REST – Core Principles

REST – Core Principles



REST – Where/How: Simple Example

REST – Where/How: Simple Example

Premise:

Data in a table could be a resource we want to read

Database server called *bbddb01*

Database called *northwind*

Table called *users*

- **http://bbddb01/northwind/users**

What, What, What?

- What type of content you return is up to you.
- Compare to SOAP where you must return XML.
- Most common are XML or JSON. You could return complex types like a picture.

REST – Is it used?

REST – Is it used?

- Web sites are RESTful

REST – Is it used?

- Web sites are RESTful
- RSS is RESTful

REST – Is it used?

- Web sites are RESTful
- RSS is RESTful
- Twitter, Flickr and Amazon expose data using REST

REST – Is it used?

- Web sites are RESTful
- RSS is RESTful
- Twitter, Flickr and Amazon expose data using REST

REST – Is it used?

- Web sites are RESTful
- RSS is RESTful
- Twitter, Flickr and Amazon expose data using REST

- Some things are “accidentally RESTful” in that they offer limited support.

Real Life: Flickr API

- Resource: Photos
- Where:
  - http://farm{farm-id}.static.flickr.com/{server-id}/{id}_{secret}.jpg
  - http://farm{farm-id}.static.flickr.com/{server-id}/{id}_{secret}_[mstb].jpg
  - http://farm{farm-id}.static.flickr.com/{server-id}/{id}_{o-secret}_o.(jpg|gif|png)
- What: JPEG, GIF or PNG (defined in the URL)
- **http://farm1.static.flickr.com/2/1418878_1e92283336_m.jpg**

REST – Methods

HTTP Methods are a key corner stone in REST.
They define the action to be taken with a URL.
Proper RESTful services expose all four – “accidental” expose less.
Nothing stopping you doing some Mix & Match
❖Some URL’s offering all of them and others a limited set

What are the four methods and what should they do?

REST – Methods

HTTP Methods are a key corner stone in REST.
They define the action to be taken with a URL.
Proper RESTful services expose all four – “accidental” expose less.
Nothing stopping you doing some Mix & Match
❖Some URL’s offering all of them and others a limited set

What are the four methods and what should they do?

REST	CRUD (Create, Read, Update, Delete)
POST	***Create***
GET	***Read***
PUT	***Update or Create***
DELETE	***Delete***

REST – Methods Example

REST – Methods Example

http://bbddb01/northwind/users[firstname=“rob%”]

REST – Methods Example

http://bbddb01/northwind/users[firstname="rob%"]

- **+ POST = Error**
- **+ GET = Returns everyone who begins with rob**
- **+ PUT = Error**
- **+ DELETE = Deletes everyone who begins with rob**

REST – Methods Example

http://bbddb01/northwind/users[firstname="rob%"]

- **+ POST = Error**
- **+ GET = Returns everyone who begins with rob**
- **+ PUT = Error**
- **+ DELETE = Deletes everyone who begins with rob**

http://bbddb01/northwind/users

& we add some input data

REST – Methods Example

http://bbddb01/northwind/users[firstname="rob%"]

- **+ POST = Error**
- **+ GET = Returns everyone who begins with rob**
- **+ PUT = Error**
- **+ DELETE = Deletes everyone who begins with rob**

http://bbddb01/northwind/users

& we add some input data

- **+ POST = Creates a new user**
- **+ GET = Returns everyone who meets criteria**
- **+ PUT = Creates/Updates a user (based on data)**
- **+ DELETE = Deletes everyone who meets criteria**

REST – Methods Example

http://bbddb01/northwind/users[firstname="rob%"]

+ POST = Error

+ PUT = Error

REST – Methods Example

http://bbddb01/northwind/users[firstname="rob%"]

+ POST = Error

+ PUT = Error

What would the error be?

HTTP 400 or 500 errors are normally used to indicate problems – same as websites

REST – Commands

You can associate commands with a resource.

Commands can replace the need for using HTTP methods and can provide a more familiar way of dealing with data.

Example:

```
userResource = new Resource('http://example.com/users/001')
userResource.delete()
```

Comparison:
REST vs. SOAP

Comparing apples and oranges

REST vs. SOAP – pt I: Technology

REST	SOAP
A STYLE	***A Standard***
Proper REST: Transport must be HTTP/HTTPS	***Normally transport is HTTP/HTTPS but can be something else***
Response data is normally transmitted as XML, can be something else. ***❖On average the lighter of the two as does not have SOAP header overhead***	***Response data is transmitted as XML***
Request is transmitted as URI ***❖Exceptionally light compared to web services*** ***❖Limit on how long it can be*** ***❖Can use input fields***	***Request is transmitted as XML***
Analysis of method and URI indicate intent	***Must analyse message payload to understand intent***
…	***WS* initiatives to improve problems like compression or security***

REST vs. SOAP – pt II: Languages

REST	SOAP
Easy to be called from JavaScript	*JavaScript can call SOAP but it is hard, and not very elegant.*
If JSON is returned it is very powerful (keep this in mind)	*JavaScript parsing XML is slow and the methods differ from browser to browser.*
C# (Visual Studio) parsing of REST means using HttpWebRequest and parsing the results (string/xml) or normal service consumption (.NET 3.5 SP 1 and later).	*C# (Visual Studio) makes consuming SOAP very easy and provides nice object models to work with.*
C# version 4 should make this easier thanks to new dynamic methods.	*...*
There are 3 with C# so that may make it easier.	*...*

REST vs. SOAP – pt III: Tools

REST	SOAP
Basic support for REST in BizTalk	*BizTalk and SOAP are made to be together.*
WCF can consume REST.	*WCF can consume SOAP.*
WCF can serve REST with a bit of tweaking.	*WCF can server SOAP.*
The new routing feature in ASP.NET 3.5 SP1 makes building a RESTful service easy.	*...*

FAQ about Security?

- Are RESTful services secure?

FAQ about Security?

- Are RESTful services secure?
  - It’s a style, not a technology so that depends on how you implement it.

FAQ about Security?

- Are RESTful services secure?
  - It's a style, not a technology so that depends on how you implement it.
- Are you open to SQL injection attacks?

FAQ about Security?

- Are RESTful services secure?
  - It's a style, not a technology so that depends on how you implement it.
- Are you open to SQL injection attacks?
- When you look at *http://bbddb01/northwind/users[firstname="rob%"]*, you may think so but you shouldn't be. Because:

1) The parameter shouldn't be SQL
2) If it is SQL, why are you not filtering it?
3) Remember the old rule: Do not trust user input

FAQ about Security?

- How can I do authentication?

FAQ about Security?

- How can I do authentication?
  - It's built on HTTP, so everything you have for authentication in HTTP is available
  - PLUS
  - You could encode your authentication requirements into the input fields

JSON

JSON – What is it?

JSON – What is it?

- *"JSON (JavaScript Object Notation) is a **lightweight data-interchange format**. It is easy for humans to read and write. It is easy for machines to parse and generate"* – JSON.org

JSON – What is it?

- *"JSON (JavaScript Object Notation) is a **lightweight data-interchange format**. It is easy for humans to read and write. It is easy for machines to parse and generate"* – JSON.org

- Importantly: JSON is a subset of JavaScript

JSON – What does it look like?

```
{
    "firstName": "John",
    "lastName": "Smith",
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": 10021
    },
    "phoneNumbers": [
        "212 555-1234",
        "646 555-4567"
    ]
}
```

JSON – What does it look like?

```
{
    "firstName": "John",
    "lastName": "Smith",
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": 10021
    },
    "phoneNumbers": [
        "212 555-1234",
        "646 555-4567"
    ]
}
```

Name/Value Pairs

Child properties

String Array

Number data type

JSON vs. XML

JSON	XML
Data Structure	*Data Structure*
No validation system	*XSD*
No namespaces	*Has namespaces (can use multiples)*
Parsing is just an eval *•Fast*	*Parsing requires XML document parsing using things like XPath*
In JavaScript you can work with objects – runtime evaluation of types	*In JavaScript you can work with strings – may require additional parsing*
Security: Eval() means that if the source is not trusted anything could be put into it.	*Security: XML is text/parsing – not code execution.*

JSON vs. XML which to use?

- Scenario 1: You have a website (say Twitter.com) and you want to expose a public API to build apps.

Issue	JSON	XML
The public will be parsing data in. You must make it secure.	***Run checks against the data in the object to make sure it's secure. You are working on objects so you must also check for potential code access issues.***	***Run checks against the data to make sure it's secure.***
Data must be in a specific format.	***Build something that parses the objects.***	***XML Schema***

JSON vs. XML which to use?

- Scenario 2: You have a website (say gmail.com) and your front end needs to show entries from a mailbox, but needs to be dynamic and so you will use a lot of JavaScript.

Issue	JSON	XML
Your in house developers know objects and would like to use them.	***JSON is JavaScript objects.***	***Write JavaScript to parse the XML to objects.***
The site is secure but you worry about people	***You page has JavaScript in it and (maybe) code which communicates with a private***	***You page has JavaScript in it and (maybe) code which communicates with a***

JSON vs. XML

JSON vs. XML

- **Which of them should you use?**
  - Use Both – They both have strengths and weaknesses and you need to identify when one is stronger than the other.

References

- Robert MacLean: JSON and REST
  - http://www.slideshare.net/rmaclean/json-and-rest
- Brian Mulloy: RESTful API design
  - http://www.slideshare.net/apigee/restful-api-design-second-edition
- Christopher Bartling et al: RESTful web services
  - http://www.slideshare.net/cebartling/rest-web-services

Using Java to implement REST Web Services: JAX-RS

Web Technology
2II25

Dr. Katrien Verbert
Dr. ir. Natasha Stash
Dr. George Fletcher

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Restful Web Services Frameworks and APIs

- JAX-RS - The Java API for RESTful Web Services
- uses annotations to make plain old Java objects (POJOs) and resources available through HTTP
- Sun Reference Project: Jersey
- Other Vendors: CXF (Apache), RESTEasy(JBoss) and Restlet

- JAX-RS tutorial:

  http://docs.oracle.com/javaee/6/tutorial/doc/gilik.html

  Source: http://www.slideshare.net/kverbert/using-java-to-implement-rest-web-services-jaxrs

JAX-RS HTTP

- Templated mapping and subresources
  - @Path
- MIME handling
  - @Provides, @Consumes
- HTTP methods
  - @GET, @POST, @UPDATE, @DELETE, @HEAD, @OPTIONS, @HttpMethod
- Caching
  - evaluatePreconditions

Example

```
package com.sun.jersey.samples.helloworld.resources;
import javax.ws.rs.GET;
import javax.ws.rs.Produces;
import javax.ws.rs.Path;

// The Java class will be hosted at the URI path "/helloworld"
@Path("/helloworld")
public class HelloWorldResource {

    // The Java method will process HTTP GET requests
    @GET
    // The Java method will produce content identified by the MIME Media
    // type "text/plain"
    @Produces("text/plain")
    public String getClichedMessage() {
        // Return some cliched textual content
        return "Hello World";
    }
}
```

@Path Annotation and URI Path Templates

@Path annotation

- identifies the URI path template to which the resource responds
- is specified at the class or method level of a resource

URI path templates are URIs with variables embedded within the URI syntax

- these variables are substituted at runtime
- variables are denoted by braces ({ and })

```
@Path("/users/{username}")
```

- example request

http://example.com/users/Galileo

@PathParam annotation

To obtain the value of the user name, the @PathParam annotation may be used on the method parameter of a request method

```
@Path("/users/{username}")
public class UserResource {

    @GET
    @Produces("text/xml")
    public String getUser(@PathParam("username") String userName) {
        ...
    }
}
```

Examples of URI Path Templates

URI Path Template	URI After Substitution
http://example.com/{name1}/{name2}/	http://example.com/james/gatz/
http://example.com/{question}/ {question}/{question}/	http://example.com/why/why/why/
http://example.com/maps/{location}	http://example.com/maps/Main %20Street

@Produces Annotation

@Produces annotation is used to specify the MIME media types or representations a resource can produce and send back to the client

- applied at the class level: default for all methods
- applied at the method level overrides any @Produces annotations applied at the class level

@Produces annotation

```
@Path("/myResource")
@Produces("text/plain")
public class SomeResource {
    @GET
    public String doGetAsPlainText() {
        ...
    }

    @GET
    @Produces("text/html")
    public String doGetAsHtml() {
        ...
    }
}
```

@Consumes Annotation

@Consumes annotation is used to specify which MIME media types of representations a resource can accept

```
@POST
@Consumes("text/plain")
public void postClichedMessage(String message) {
    // Store the message
}
```

Using Java to implement SOAP web Services: JAX-WS

Web Technology
2II25

Dr. Katrien Verbert
Dr. ir. Natasha Stash
Dr. George Fletcher

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

JAX-WS 2.0

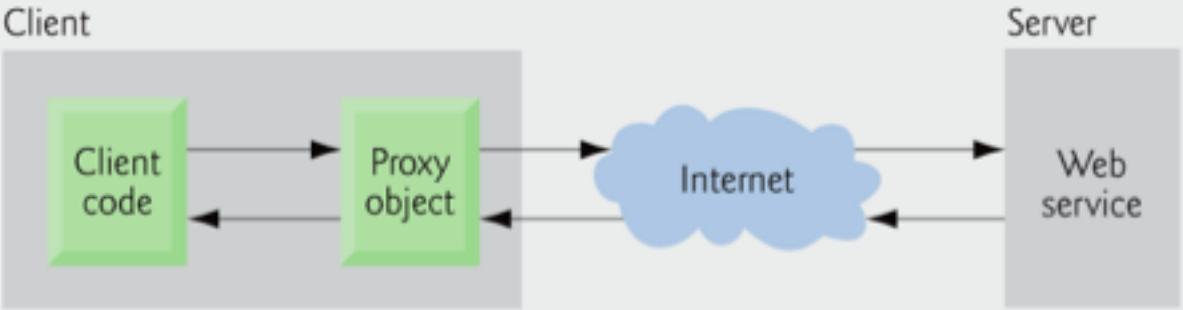
- Part of Java EE
- New in Java SE 6
- API stack for web services.
- New API's:
  - JAX-WS, SAAJ, Web Service metadata
- New packages:
  - javax.xml.ws, javax.xml.soap, javax.jws

Java Web Services Basics

- Remote machine or server
  - The computer on which a web service resides
- A client application that accesses a web service sends a method call over a network to the remote machine, which processes the call and returns a response over the network to the application
- In Java, a web service is implemented as a class that resides on a server
- Publishing a web service
  - Making a web service available to receive client requests
- Consuming a web service
  - Using a web service from a client application

Communication between JAX-WS Web Service and Client

Java Web Services Basics



Writing a webservice

```
package loanservice;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.xml.ws.Endpoint;

@WebService
public class LoanApprover {

    @WebMethod
    public boolean approve(String name) {
        return name.equals("Mike");
     }
}
```

Annotations

Annotations are a new feature of JDK 1.5 and later.

- Essentially they are markers in the Java source code
- That can be used by external tools to generate code

Format looks like

```
@ThisIsAnAnnotation(foo="bar")
```

Annotations can occur only in specific places in the code

- before a class definition,
- before a method declaration, ...

Requirements of a JAX-WS Endpoint

Requirements of a JAX-WS Endpoint

- The implementing class must be annotated with the @WebService or @WebServiceProvider annotation

Requirements of a JAX-WS Endpoint

- The implementing class must be annotated with the @WebService or @WebServiceProvider annotation
- The business methods of the implementing class must be public.

Requirements of a JAX-WS Endpoint

- The implementing class must be annotated with the @WebService or @WebServiceProvider annotation
- The business methods of the implementing class must be public.
- The business methods must not be declared static or final.

Requirements of a JAX-WS Endpoint

- The implementing class must be annotated with the @WebService or @WebServiceProvider annotation
- The business methods of the implementing class must be public.
- The business methods must not be declared static or final.
- Business methods that are exposed to web service clients must be annotated with @WebMethod.

Requirements of a JAX-WS Endpoint

- The implementing class must be annotated with the @WebService or @WebServiceProvider annotation
- The business methods of the implementing class must be public.
- The business methods must not be declared static or final.
- Business methods that are exposed to web service clients must be annotated with @WebMethod.
- Business methods that are exposed to web service clients must have JAXB-compatible parameters and return types.
    - See the list of JAXB default data type bindings at
    - http://docs.oracle.com/javaee/5/tutorial/doc/bnazq.html#bnazs.

@WebService annotation

- Indicates that a class represents a web service
- Optional element name
  - specifies the name of the proxy class that will be generated for the client
- Optional element serviceName
  - specifies the name of the class to obtain a proxy object.

Creating, Publishing, Testing and Describing a Web Service

Calculator web service

- Provides method that takes two integers
- Can determine their sum

CalculatorWS example

```
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@WebService(serviceName = "CalculatorWS")
public class CalculatorWS {

    @WebMethod
    public int add (@WebParam (name= "value1") int value1,
      @WebParam( name="value2" ) int value2){
        return value1 + value2;
    }
}
```

Declare that method add is a WebMethod

Specify parameter names

Coding the Service Endpoint Implementation Class

- @WebService annotation at the beginning of each new web service class you create
- @WebMethod annotation at the beginning of each method that is exposed as a WSDL operation
  - Methods that are tagged with the @WebMethod annotation can be called remotely
  - Methods that are not tagged with @WebMethod are not accessible to clients that consume the web service
- @WebParam annotation is used here to control the name of a parameter in the WSDL
  - Without this annotation the parameter name = arg0

- @WebMethod annotation
  - Optional operationName element to specify the method name that is exposed to the web service's client
- Parameters of web methods are annotated with the @WebParam annotation
  - Optional element name indicates the parameter name that is exposed to the web service's clients

Building, Packaging, and Deploying the Service

Java IDEs

- Netbeans
  - download: http://netbeans.org/
  - tutorial: http://netbeans.org/kb/docs/websvc/jax-ws.html?print=yes
- Eclipse
  - download: http://www.eclipse.org/
  - tutorial: http://rantincsharp.wordpress.com/2008/10/14/a-simple-soap-web-service-example-in-eclipse-ganymede/
- IntelliJ IDEA
  - download: http://www.jetbrains.com/idea/
  - tutorial: http://wiki.jetbrains.net/intellij/Web_Services_with_IntelliJ_IDEA#JAX_WS

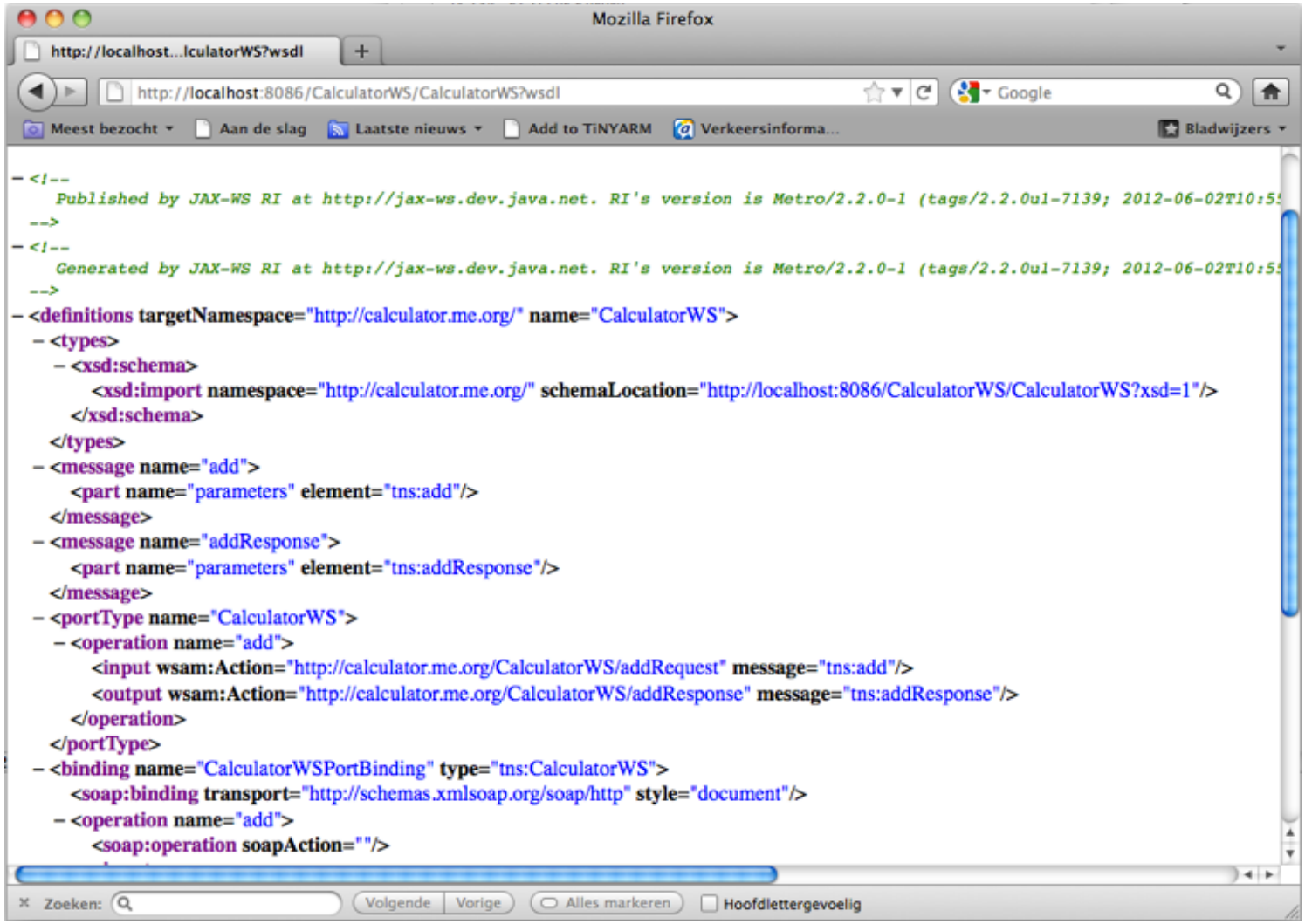
Using Ant

http://docs.oracle.com/javaee/6/tutorial/doc/bnayn.html

Describing a Web Service with the Web Service Description Language (WSDL)

- To consume a web service
  - Must know where to find the web service
  - Must be provided with the web service's description
- Web Service Description Language (WSDL)
  - Describe web services in a platform-independent manner
  - The server generates a WSDL dynamically for you
  - Client tools parse the WSDL to create the client-side proxy class that accesses the web service
- To view the WSDL for a web service
  - Type URL in the browser's address field followed by ?WSDL or
  - Click the WSDL File link in the Sun Java System Application Server's Tester web page

Example WSDL



Creating a Client in Netbeans to Consume a Web Service

- When you add a web service reference
  - IDE creates and compiles the client-side artifacts
  - the framework of Java code that supports the client-side proxy class
- Client calls methods on a proxy object
  - Proxy uses client-side artifacts to interact with the web service
- To add a web service reference
  - Right click the client project name in the Netbeans Projects tab
  - Select New > Web Service Client…
  - Specify the URL of the web service's WSDL in the dialog's WSDL URL field

Calculator client

```
import calculator.*;

public class CalculatorClient {
    public static void main(String[] args) {
        CalculatorWS_Service service=new CalculatorWS_Service();
        CalculatorWS port= service.getCalculatorWSPort();
        int result = port.add(2, 3);
        System.out.println(result);
    }
}
```

Relevant links

- **Netbeans tutorial for developing a SOAP-based web services:**

  http://netbeans.org/kb/docs/websvc/jax-ws.html

- **Building SOAP-based web services with JAX-WS:**

  http://docs.oracle.com/javaee/6/tutorial/ doc/ bnayl.html

Source: http://www.slideshare.net/kverbert/soap-tutorial

SOAP and XML processing

Web Technology
2II25

Dr. Katrien Verbert
Dr. ir. Natasha Stash
Dr. George Fletcher

http://www.slideshare.net/kverbert/soap-tutorial

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

XML document

```
<?xml version="1.0"?>
  <Order>
        <Date>2003/07/04</Date>
        <CustomerId>123</CustomerId>
        <CustomerName>Acme Alpha</CustomerName>
        <Item>
            <ItemId> 987</ItemId>
            <ItemName>Coupler</ItemName>
            <Quantity>5</Quantity>
        </Item>
        <Item>
            <ItemId>654</ItemId>
            <ItemName>Connector</ItemName>
            <Quantity unit="12">3</Quantity>
        </Item>
   </Order>
```

Parsing XML

Goal

Read XML files into data structures in programming languages

Possible strategies

- Parse into generic tree structure (DOM)
- Parse as sequence of events (SAX)
- Automatically parse to language-specific objects (JAXB)

JAXB

JAXB: Java API for XML Bindings

Defines an API for automatically representing XML schema as collections of Java classes.

Most convenient for application programming

Annotations markup

@XmlAttribute to designate a field as an attribute

@XmlRootElement to designate the document root element.

@XmlElement to designate a field as a node element

@XmlElementWrapper to specify the element that encloses a repeating series of elements

Note that you should specify only the getter method as @XmlAttribute or @XmlElement.

Jaxb oddly treats both the field and the getter method as independent entities

Order example

```
import javax.xml.bind.annotation.*;

@XmlRootElement
public class Item {

    @XmlElement
    private String itemId;
    @XmlElement
    private String ItemName;
    @XmlElement
    private int quantity;

    public Item() {
    }
}

}
```

Order example

```
import javax.xml.bind.annotation.*;
import java.util.*;

@XmlRootElement
public class Order {

    @XmlElement
    private String date;
    @XmlElement
    private String customerId;
    @XmlElement
    private String customerName;
    @XmlElement
    private List<Item> items;

    public Order() {
        this.items=new ArrayList<Item>();
    }


}
```

Marshalling

marshalling

the process of producing an XML document from Java objects

unmarshalling

the process of producing a content tree from an XML document

JAXB only allows you to unmarshal valid XML documents

JAXB only allows you to marshal valid content trees into XML

Marshalling example

```
public String toXmlString(){
        try{
            JAXBContext context=JAXBContext.newInstance(Order.class);
            Marshaller m = context.createMarshaller();
            m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE);
            ByteArrayOutputStream b=new ByteArrayOutputStream();
            m.marshal(this,b);
            return b.toString();
        }catch (Exception e){
            e.printStackTrace();
            return null;
        }
}
```

Unmarshalling example

```
public Order fromXmlString(String s){
        try{
            JAXBContext jaxbContext = JAXBContext.newInstance(Order.class);
            Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller()
            Order order = (Order) jaxbUnmarshaller.unmarshal(new StreamSource( new
             StringReader(s)));
            return order;
        }catch (Exception e){
            e.printStackTrace();
            return null;
        }
}
```

Test transformation

```
public static void main(String args[]){
        Order o=new Order("1 March 2013", "123", "Katrien");
        o.getItems().add(new Item("1", "iPhone 5", 2));
        o.getItems().add(new Item("2", "Nokia Lumia 800", 2));
        System.out.println(o.toXmlString());

    }
```

Output

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<order>
   <customerId>123</customerId>
   <customerName>Katrien Verbert</customerName>
   <date>12 February 2013</date>
   <items>
      <itemId>id1</itemId>
      <ItemName>Iphone 5</ItemName>
      <quantity>2</quantity>
   </items>
   <items>
      <itemId>id2</itemId>
      <ItemName>Nokia Lumia 800</ItemName>
      <quantity>1</quantity>
   </items>
</order>
```

Developing Cloud Applications with Eclipse Gyrex

Gunnar Wagenknecht, @guw

http://www.eclipsecon.org/europe2012/sessions/developing-cloud-applications-eclipse-gyrex

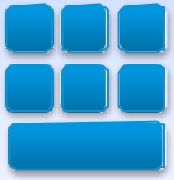
Modern Server Applications

- **High traffic**

Million transactions per hour

- **Different** frontend **technologies** and **devices**

- **Modular** in development and deployment



- **Easy** to setup and run

- **Open** for new technologies
  - e.g. persistence

Eclipse Gyrex

A lightweight **application stack** for building server applications using **EclipseRT** technologies.

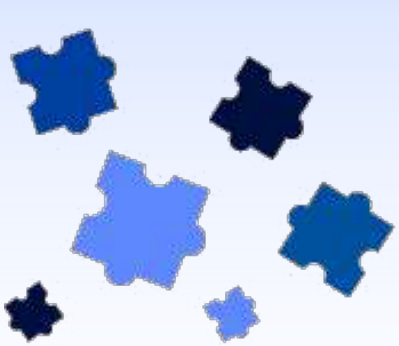
EclipseRT (RT = Runtime)

"EclipseRT is the collection of OSGi-based runtimes and frameworks built by the Eclipse open source projects. "

Containers, Middleware, EnterpriseFrameworks

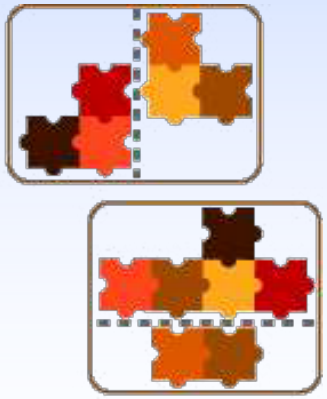
Equinox

- **OSGi** reference implementation
- Foundation of EclipseRT
- **C**omponent **O**riented **D**evelopment and **A**ssembly



Create

Extend



Assemble

equinox
OSGi

Jetty

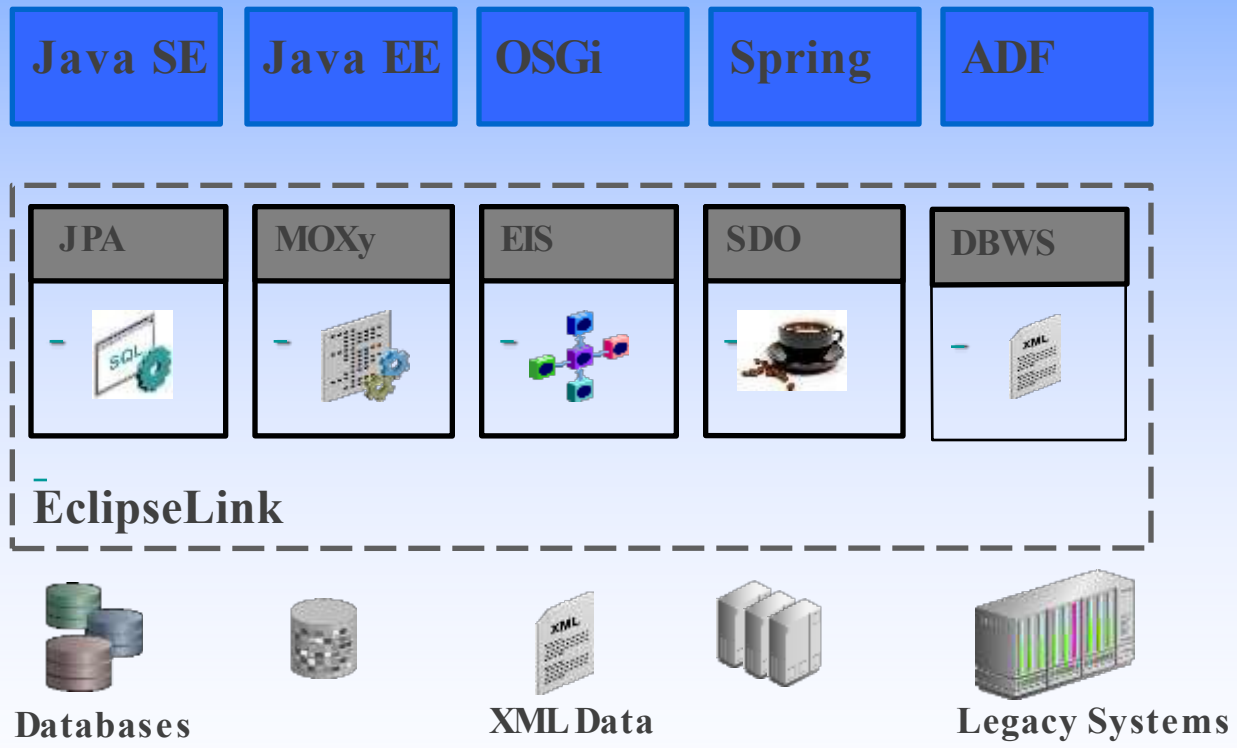
- Asynchronous HTTP Server and Client
- Standards based Servlet Container
- Web Sockets server
- OSGi, JNDI, JMX, JASPI, AJP support

- Small foot print
- Excellent scalability

- Runs in
  - Apache Hadoop
  - Google AppEngine

EclipseLink

Comprehensive Java persistence solution addressing relational, XML, and database web services.



eclipse link

Gyrex

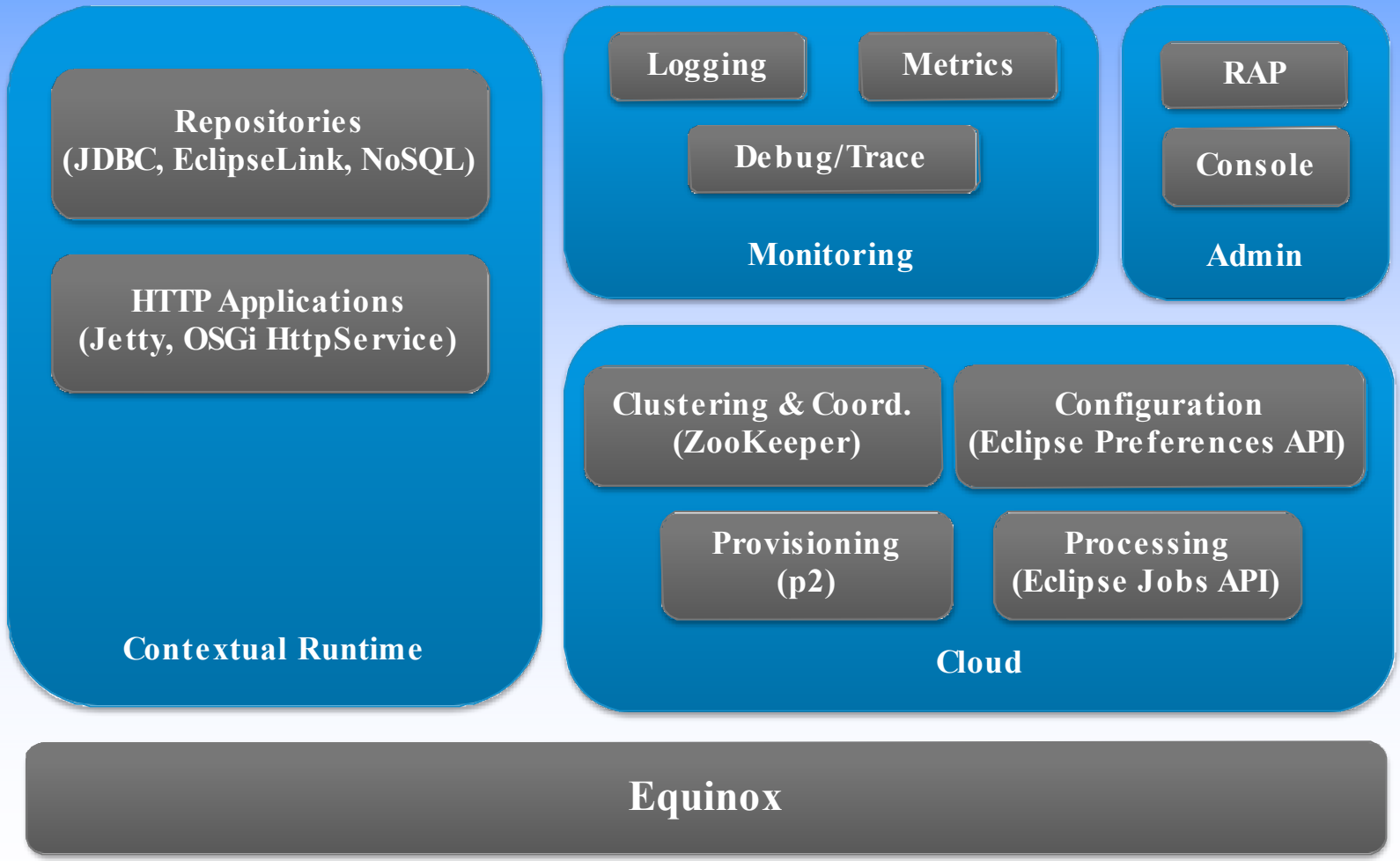
- built-in **clustering**
- built-in **web-based administration** UI
- built-in **multi tenancy**
- enhancements for **professional maintenance**
  - centralized logging
  - cluster provisioning UI

GYREX

Gyrex Features

- lightweight **application stack**
- fast **100% OSGi** runtime
- central **cluster** configuration through Apache ZooKeeper
- cluster aware **job scheduling**
- **automated deployment** through p2
- support for cluster **node roles**, e.g. “job worker node” and “api node”

Gyrex Components



Gyrex Infrastructure Setup

For a high traffic application with different frontends

GYREX

iPad App
Mobile Apps

PHP Web App

Internet Of Things Device

Native App

Load Balancer

REST API

Gyrex Node

p2 repo

data repositories

ZK Node

ZooKeeper Cluster

Coordinates the cluster and holds the complete **application-** and **cluster configuration**

Q&A

- Gyrex Newsgroup / Forum at
  http://www.eclipse.org/forums/

- Information hub at
  http://www.eclipse.org/gyrex/

- Session feedback / questions
  gunnar@eclipse.org
  @guw