

# TDP024

Enterprise Systems

Introduktion, ES, SOA, Multi-Tier

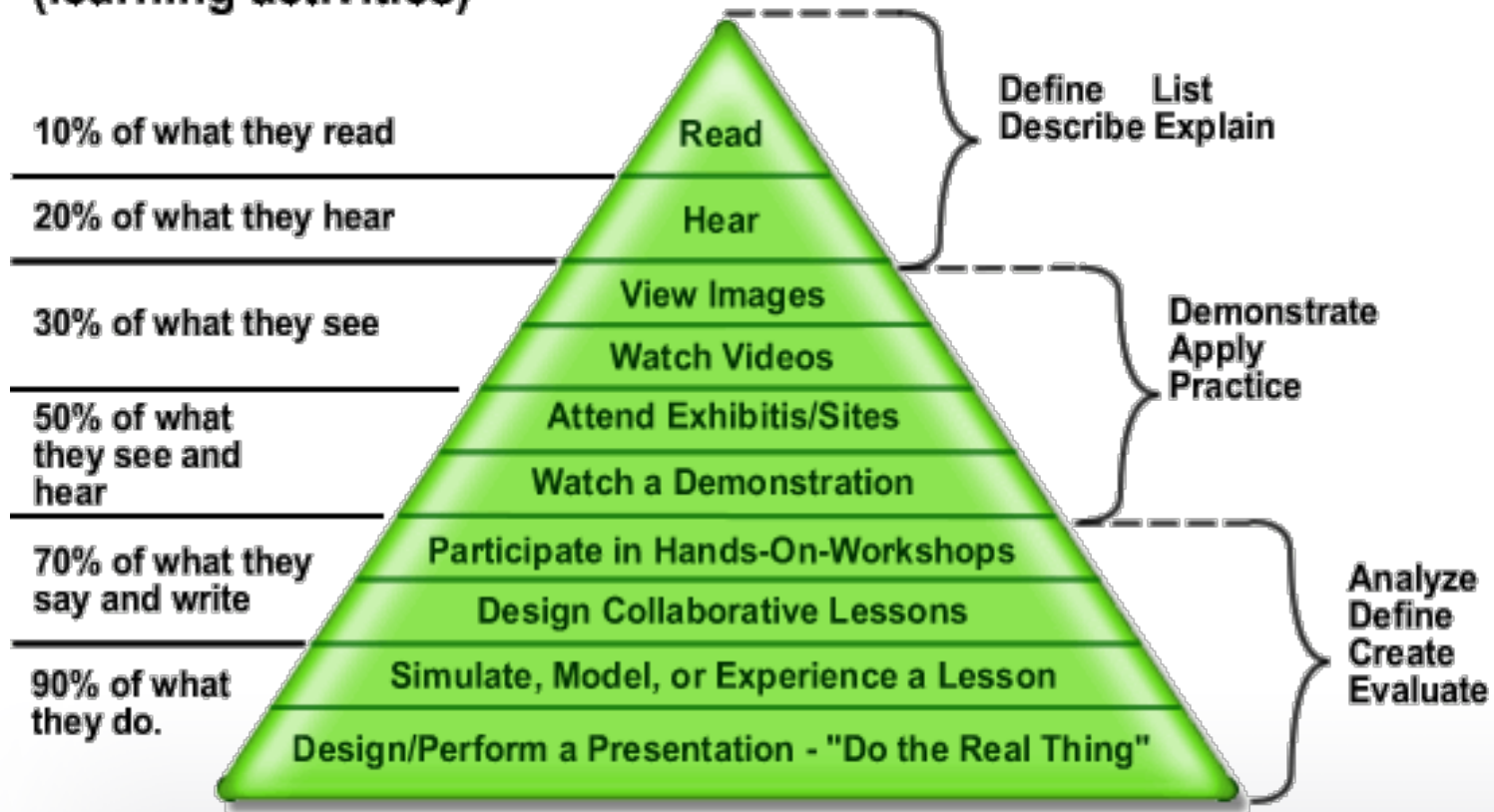
Anders Fröberg ([anders.froberg@liu.se](mailto:anders.froberg@liu.se))

Institutionen för Datavetenskap (IDA)

# Kursens kunskapsfilosofi

People generally remember...  
(learning activities)

People are able to...  
(learning outcomes)



# KOMMER NI KOMMA IHÅG ?

- Load up your guns, and bring your friends
- This is the end the only end my Friend
- Äntligen stod prästen i predikstolen
- I'll be back
- Han kom som ett yrväder en aprilafton och hade ett höganäskrus i en svångrem om halsen
- Jag har inget fint att ge, mer än 60 kilo ensamhet
- It was the best of times, it was the worst of times
- Face it tiger, you just hit the jackpot!

# TDP024 – Kursinformation

- Kurshemsidan innehåller information om laborationerna, seminariet och examinationen.
- Föreläsningskoden finns på hemsidan, detta är kurslitteraturen för denna kurs. **Att förstå hur denna kod är uppbyggd, och att följa de riktlinjer som ges i denna kod är avgörande för att få godkänt på laborationerna.**
- Kursen ger betyg U/3/4/5
- **Kursansvarig:** Anders Fröberg ([anders.froberg@liu.se](mailto:anders.froberg@liu.se))
- **Kursassistent:** Sahand Sadjadee [sahand.sadjadee@liu.se](mailto:sahand.sadjadee@liu.se)
- **Examinator:** Anders Fröberg

# TDP024 – In a nutshell

- I examensordningen står det att för alla kandidatexamina skall bland andra följande mål uppnås: (Även andra examina än kandidatexamina har nästan identiska mål)
  1. visa förmåga att **söka, samla, värdera** och **kritiskt tolka** relevant information i en problemställning samt att kritiskt diskutera företeelser, frågeställningar och situationer
  2. visa sådan färdighet som fordras för att **självständigt arbeta** inom det område som utbildningen avser
  3. visa förmåga att **självständigt identifiera, formulera och lösa problem** samt att genomföra uppgifter **inom givna tidsramar**
  4. visa förmåga att **muntligt** och **skriftligt** redogöra för och diskutera information, problem och lösningar i dialog med olika grupper

# Utvärdering och Förändringar

- Oklarheter vad som tar tid
  - Lab 3 -> Projekt
- Vad krävs för olika betyg
  - Lab 4 -> Betyg 4
- Tips från student
- Mockito

# TDP024 – In a nutshell part 2

- Laborationer +Projekt med betyg 5Hp
- Två labbar
  - Git (lab 1)
  - Maven (lab 2, lite lättare för i år)
- Projekt
  - SOA+Kafka
- Individuell rapport och seminarier 1Hp
  - Sök och välj ut tre bra forskningsartiklarna relaterade till kursens innehåll . Skriv en sammanfattning och analys av alla tre artiklarna på en A4-sida vardera.
  - Värdera varje artikel - Varför är den bra
  - Presentera och Diskutera på seminarium
- Mer kodning , skriva fler egna tjänster
  - Visa er kod för andra
- Handuppräkring i Camedin

# TDP024 – Laborationer/Scenarion

- Laborationerna genomförs i par, registrering via webreg.
- Salarna: Thinlinc/RDP till Linux maskiner, SU-sal, eller egen dator.
- Ni använder valfri IDE/editor för utveckling, Maven för att bygga



1

Kan du utveckla en iPhone app åt oss?



Absolut, inga problem alls.

iPhone app  
Swift  
If(a != b) ...  
else ....

2

Vi skulle behöva en till Android också ...

... ok ...



Android App  
Kotlin  
If(a != b) ...  
else ....

*Det börjar ta fart det här.  
Kan du göra en desktop  
applikation och en  
webbsida också ?*



*... mmm ...*

*Android App*  
**Kotlin**  
*If(a != b) ...  
else ....*

*iPhone app*  
**Swift**  
*If(a != b) ...  
else ....*

*Desktop*  
**Winforms**  
*If(a != b) ...  
else ....*

*Webbsida*  
**Python**  
*If(a != b) ...  
else ....*

*Vi har pratat med våra kunder, de tycker att "a" skall vara lika med "b"*

...

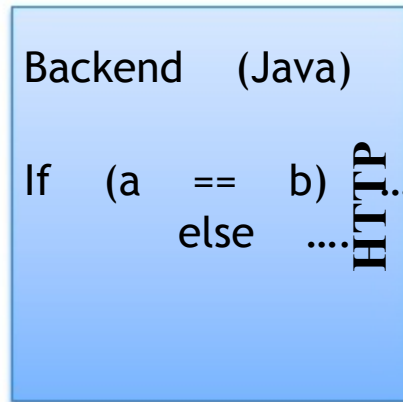


*Fyra applikationer, alla skriva i olika språk som nu skall ändra sin logik.*

*Sedan måste jag uppdatera mina applikationer till "Play" och "Appstore", webbservern skall uppdateras och alla som har installerat programmet på sin desktop måste också uppdatera ...*

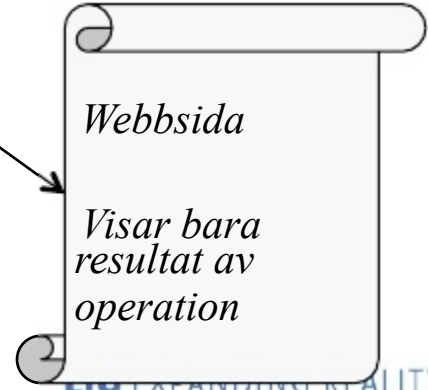
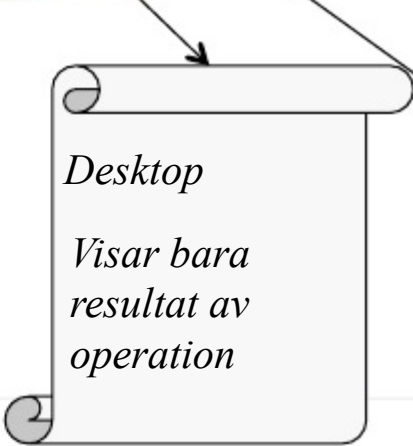
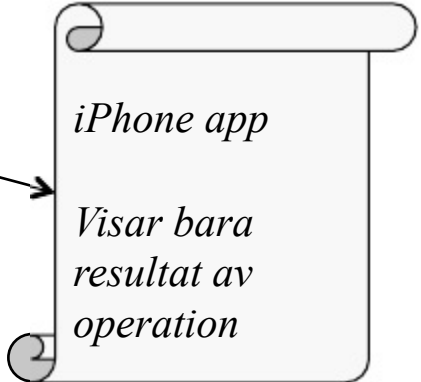
*Vad var det nu DRY stod för igen ?*

# Multi-tier – Dela upp data och vy



Ändrar på ett ställe, i ett språk och behöver inte uppdatera klienterna.

Vi ångrade oss, kan "a" vara lika med "b" istället...



Kunderna älskar applikationerna, vi har tänkt att vi ska ha försäljning i applikationerna. Fixar du kreditkortsbetalningar ?

PCI security standards är komplicerade. Det kräver att de synar mina system i sömmarna och att jag lär mig allt om deras krav.

Kanske måste jag certifiera mina applikationer varje år ?



## Företag KreditKort AB

*Ge oss 2kr för varje transaktion, så fixar vi betalningarna åt dig. Vi är redan PCI certifierade.*

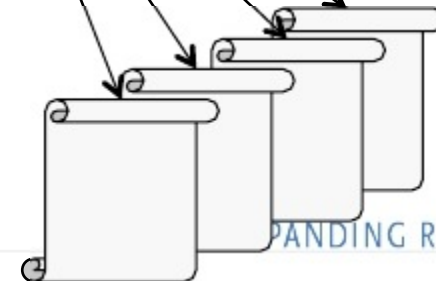


Kreditkortstjänst

HTTPS

Backend

*Taget! 2kr är det värt. Jag kopplar ihop min backend med eran tjänst.*



PANDING REALITY

*Allt fungerar superbra! Jag  
råkade köpa ett C++ bibliotek  
för 50000kr och min chef kommer  
nog inte bli glad om  
vi inte använder det.  
Kan du få in det  
någonstans i systemet?*

*Min backend är  
skriven i Java och jag  
kan väldigt lite om C++*





*Undrar vilka "dependencies"  
deras C++ bibliotek har på  
andra bibliotek, O/S, etc...*

*Det finns ju native i Java  
som kanske kan köra C++  
koden, men jag är inte  
säker hur stabilt det är.*

*Vad händer om de ändrar  
i C++ koden ?*

*Vad var det  
KISS stod för nu  
igen ?*



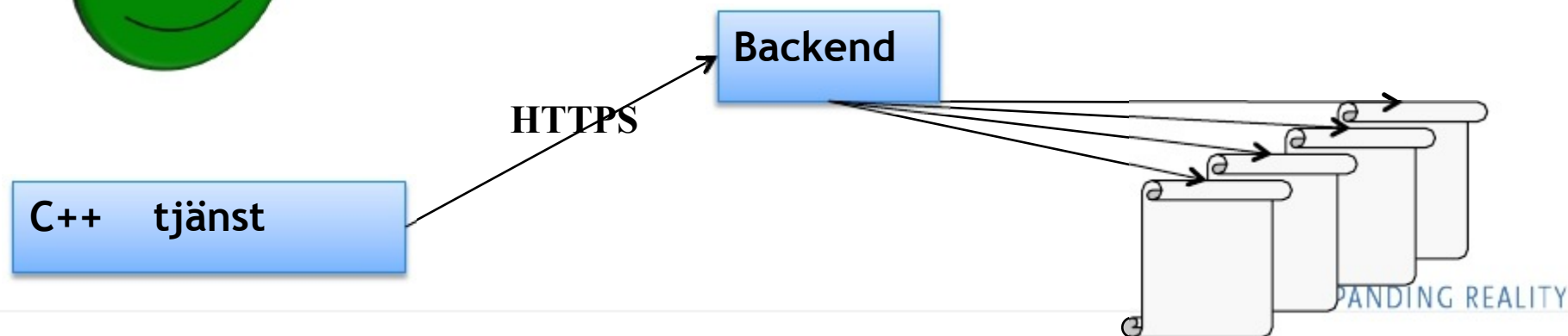


## Använd kompetensen inom företaget!

*Jag kan C++, ge mig biblioteket så skapar jag en tjänst som du kan anropa via en URL och HTTP.*

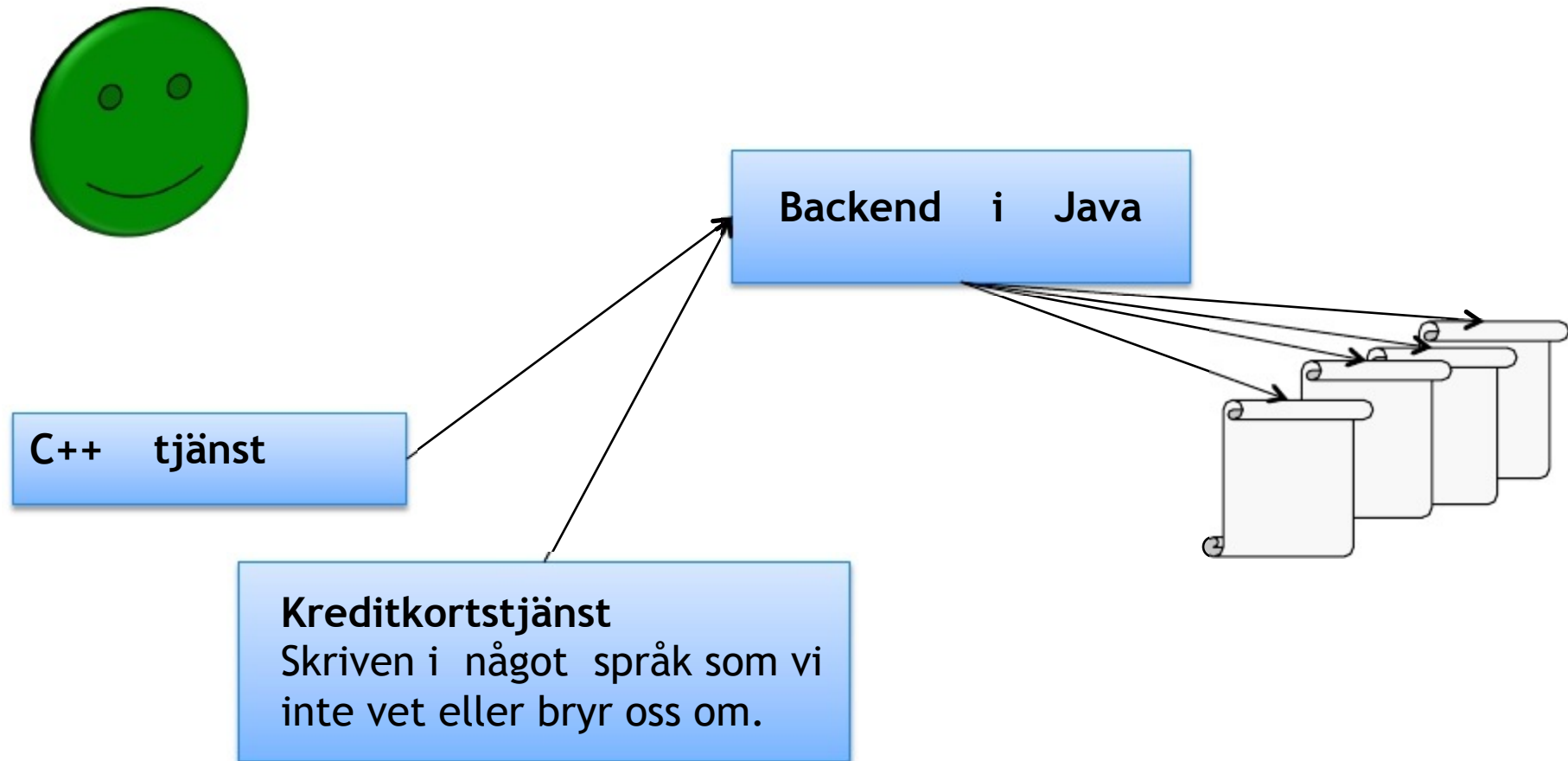


*Bra! Då slipper jag bry mig om alla detaljer som jag ändå inte kan!*



# SOA – Service Oriented Architecture

*Outsourcing av moduler inom och utom företaget*



## Intermezzo – SOA och HTTP

- Det är inte nödvändigt att använda HTTP i SOA. Man kan använda vad som helst för system för att kommunicera över nätverk.
- System som CORBA, DCOM har använts förr men visar sig vara onödigt komplicerade. De är inte heller framtidskompatibla, (CORBA i iPhone?)
- Vi använder HTTP i våra SOA tjänster, och det är det vanligaste att man gör idag.
- Dock måste man tänka på att existerande tjänster kan ha utvecklats under en period då CORBA eller DCOM var populärt.

# Intermezzo – REST och SOAP

## REST

- Vi anropar vår backend genom att konstruera URL:er

*GET example.com/account/124/  
username*

- Returvärde ofta i JSON eller XML
- Om man vill skicka med parametrar gör man det som vanligt:

*GET example.com/account?id=3*

## Old school such as SOAP

- Ett strikt protokoll där information skickas fram och tillbaka med XML

POST example.com  
.... xml dokument ...

- XML dokumentet beskriver vilken metod som skall exekveras och vilka parametrar som skickas.

# Intermezzo – REST och SOAP - Hybrid

- Det vanliga är att man blandar dessa lite, en slags hybrid.
- Amazon EC2 och Route 53 använder en hybrid där man ibland skickar skickar en begäran som liknar REST och får tillbaka XML men ibland måste författa ett XML dokument (dock inte enligt SOAP standarden).
- Twitter har något de kallar ”GET Search”, där man själv kan knopa ihop en url för att få tillbaka en twitter feed i JSON.

*<http://search.twitter.com/search.json?q=rest>*



# Intermezzo – REST och SOAP - Denna kurs

- I denna kurs använder vi REST. SOAP är onödigt komplicerat och håller på att fasas ut i många stora system.
- Vi använder parametrar in i våra tjänster och får ut JSON (som Twitter GET search).

`http://www.example.com?id=5`

`{'id':5,'content':'Hello World'}`

```
graph TD; A[http://www.example.com?id=5] --> B[Backend i Java]; B --> C[{'id':5,'content':'Hello World'}];
```

**Backend i Java**

# SOA - Principer

- SOA har inget bibliotek, API, kontrakt eller regelverk som man måste följa.
- SOA är ett sätt att dela in ett system i små delar som alla kan kommunicera.
- Hur man gör detta är upp till individen eller företaget.
- Trots detta finns det några riktlinjer, eller principer, som man bör följa för att lyckas med sin SOA.
- Notera dock att SOA tjänster har utvecklats de senaste 10-20 åren, och man kan i verkligheten stöta på äldre SOA tjänster som inte följer dessa principer.

# SOA – Principer - Kontrakt

- En tjänst behöver någon typ av dokumentation som talar om hur man anropar tjänsten.

## Request Parameters

Name	Description	Required
InstanceIds	One or more instance IDs. Type: String Default: None	Yes

## Response Elements

The elements in the following table are wrapped in a `StartInstancesResponse` structure.

Name	Description
requestId	The ID of the request. Type: xsd:string
instancesSet	List of instance state changes. Each change's information is wrapped in an <code>Item</code> element. Type: InstanceStateChangeType

## Examples

### Example Request

This example starts the `i-10a64379` instance.

```
https://ec2.amazonaws.com/?Action=StartInstances
&InstanceIds=i-10a64379
&AUTHPARAMS
```

### Example Response

```
<StartInstancesResponse xmlns="http://ec2.amazonaws.com/doc/2011-07-15/">
  <requestId>59df299-3bd4eac-93ed-be58EXAMPLE</requestId>
  <instancesSet>
    <item>
      <instanceId>i-10a64379</instanceId>
      <currentState>
        <code>0</code>
        <name>pending</name>
      </currentState>
      <previousState>
        <code>80</code>
        <name>stopped</name>
      </previousState>
    </item>
  </instancesSet>
</StartInstancesResponse>
```

<http://docs.amazonwebservices.com/AWSEC2/latest/APIReference/>

4



# SOA – Principer – Abstraction and Autonomy

- Tjänsten skall ha full kontroll över sin egen logik och hålla den gömd från klienten (gömd som i abstraherad).

*Jag älskar min nya email tjänst.  
Skickar 10000 email utan problem.*

*Jag vet inte hur det fungerar, men  
mejlen kommer fram.  
Jag knopar bara ihop en URL och  
anropar den från min tjänst.*

*Skicka 10000 email*

## SOA Tjänst Email



*Oh no, jag visste att jag  
skulle automatiserat detta.  
Nu får jag sitta hela natten  
och skicka 10 000 email.  
Men kunden märker inget  
☺☺*

# SOA – Principer - Stateless

- Tjänsten skall vara stateless

*Kräver lite fler bilder →*

Vi håller på att utöka vårt utbud, kan du skapa en "kundvagn" ?

Det finns ju Sessions i min server mjukvara, på så vis kan servern automatiskt hålla reda på vem som är vem, och jag kan lagra varor i sessionsdata



Jag vill inte spara "korgen" i databasen, då får jag massor med död data då användare fyller korgar och inte checkar ut ...

## Backend

Spara sessionsdata på servern som kommer ihåg användare mellan anrop och sparar deras kundvagn

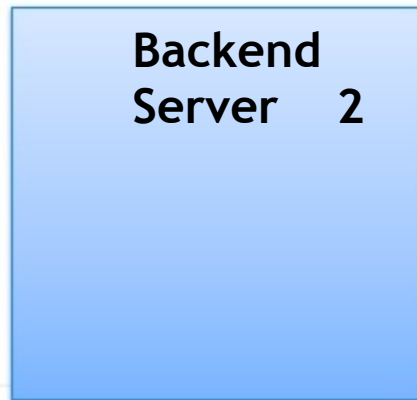
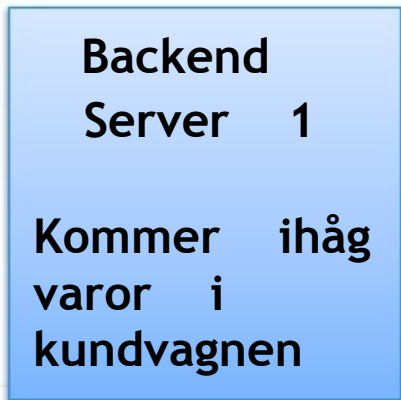
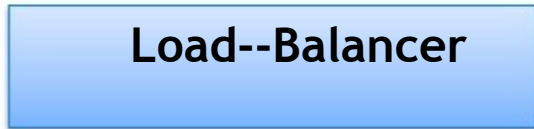
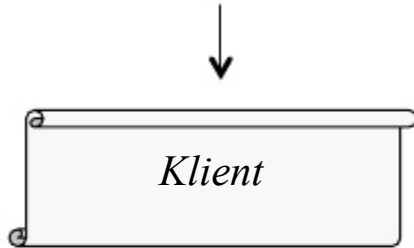
*Vi har precis skaffat pengar till en tv-reklam, räkna med 3-5 gånger så mycket trafik som vanligt!*

*Det borde inte vara något problem om jag startar några extra servrar, skalar ut, och sätter en load-balancer mellan dessa*



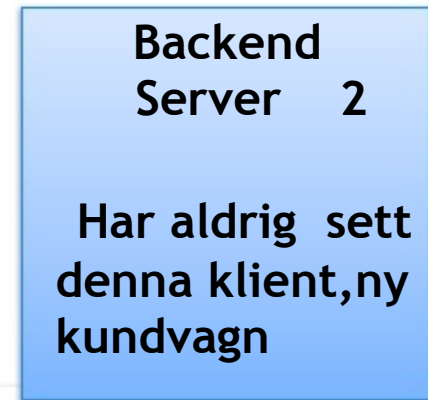
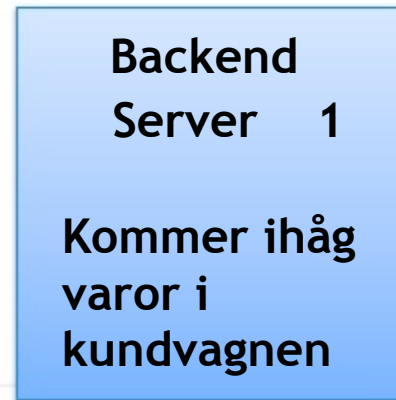
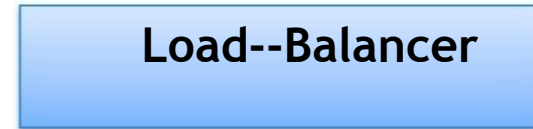
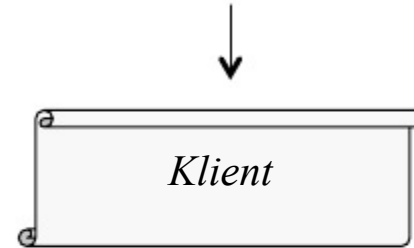
**1**

En klient lägger till en vara i sin kundvagn



**2**

En klient lägger till en vara i sin kundvagn

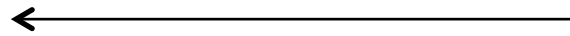


Server 1

Server 2

Load Balancer

*Skicka hela kundvagnen när det är dags att "checka ut"*



*Klienten ansvarar för att ha en "kundvagn"*



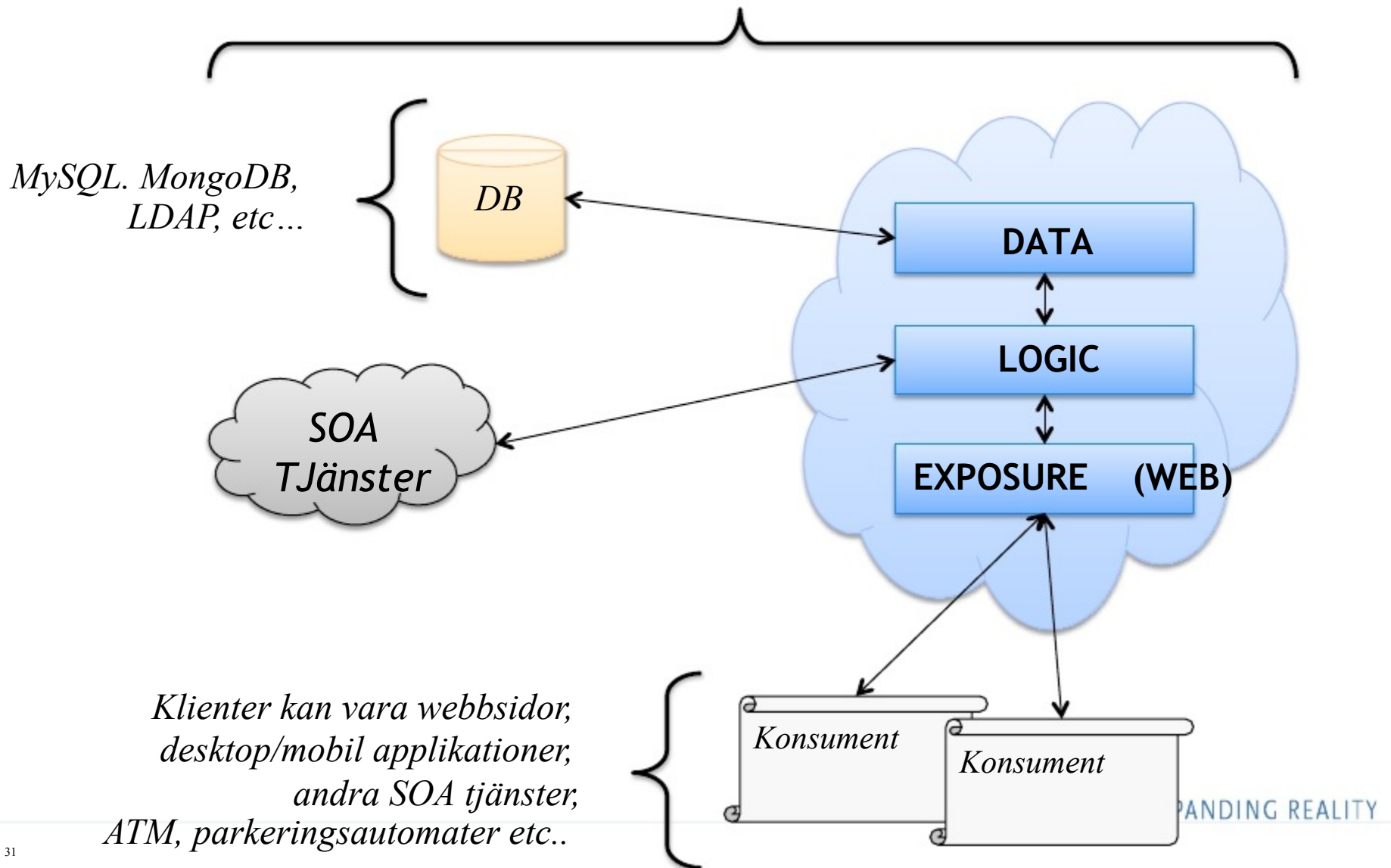
*Ser jag till att min backend inte minns varje anrop så har jag "statelessness", och då blir det lättare att skala bakom en load-balancer vid hög trafik*

# SOA Principer - Summering

1. Contract
2. Abstraction
3. Autonomy
4. Stateless

# Backend – SOA Tjänst – Egentligen flera lager

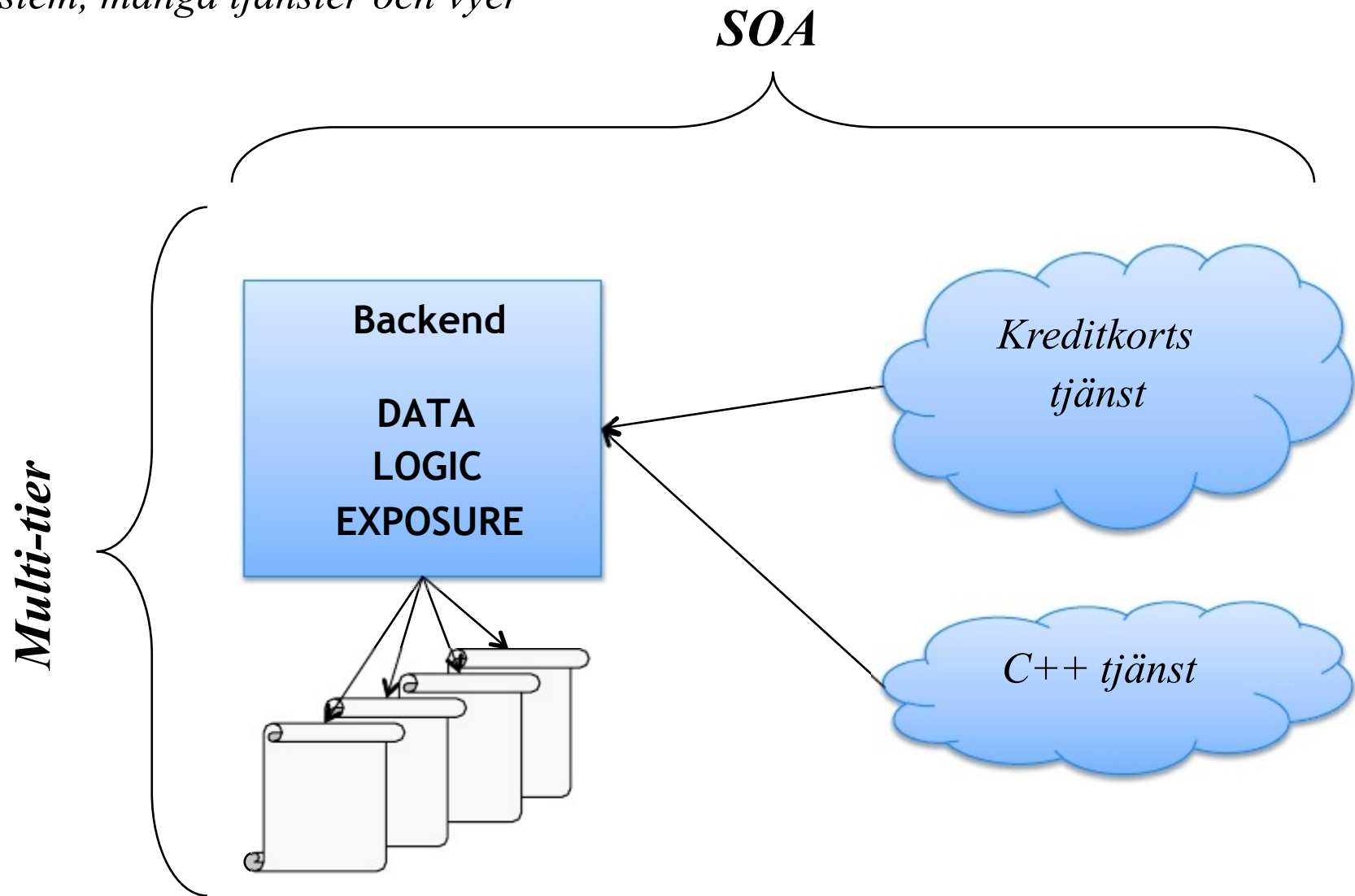
**Backend – DB, DATA, LOGIC, EXPOSURE (WEB)**





# Enterprise Systems

*Ett system, många tjänster och vyer*



# Inte bara Java

- Vi använder mycket Java i denna kurs.
- Men det går att utveckla för SOA med vilket språk som helst (som har något form av nätverksstöd).
- Det är den frihet man vill uppnå med SOA, att inte tvinga på något språk eller verktyg.
- Tänk genom hela kursen: Hur hade jag gjort detta i språk X med ramverk Y ?
- Poängen med den här kursen är att förstå arkitekturen och de problem som uppstår när man vill använda sig av just SOA.

# Varför Java ?

- Utan tvekan det mest använda språket för Enterprise Systems idag.
- Vore oansvarigt att inte ge en introduktion till Java EE som del av en komplett utbildning inom programmering och/eller datavetenskap.
- Även om Oracle äger Java och därmed Java EE så har bland andra följande företag stort inflytande över utvecklingen tack vare stora investeringar av tid, utvecklare och pengar:  
*Credit Suisse, Google, VM Ware, Eclipse Foundation, Hewlett-Packard, Ericsson, IBM, RedHat, Fujitsu, Intel, SAP*

# Java

## Java SE

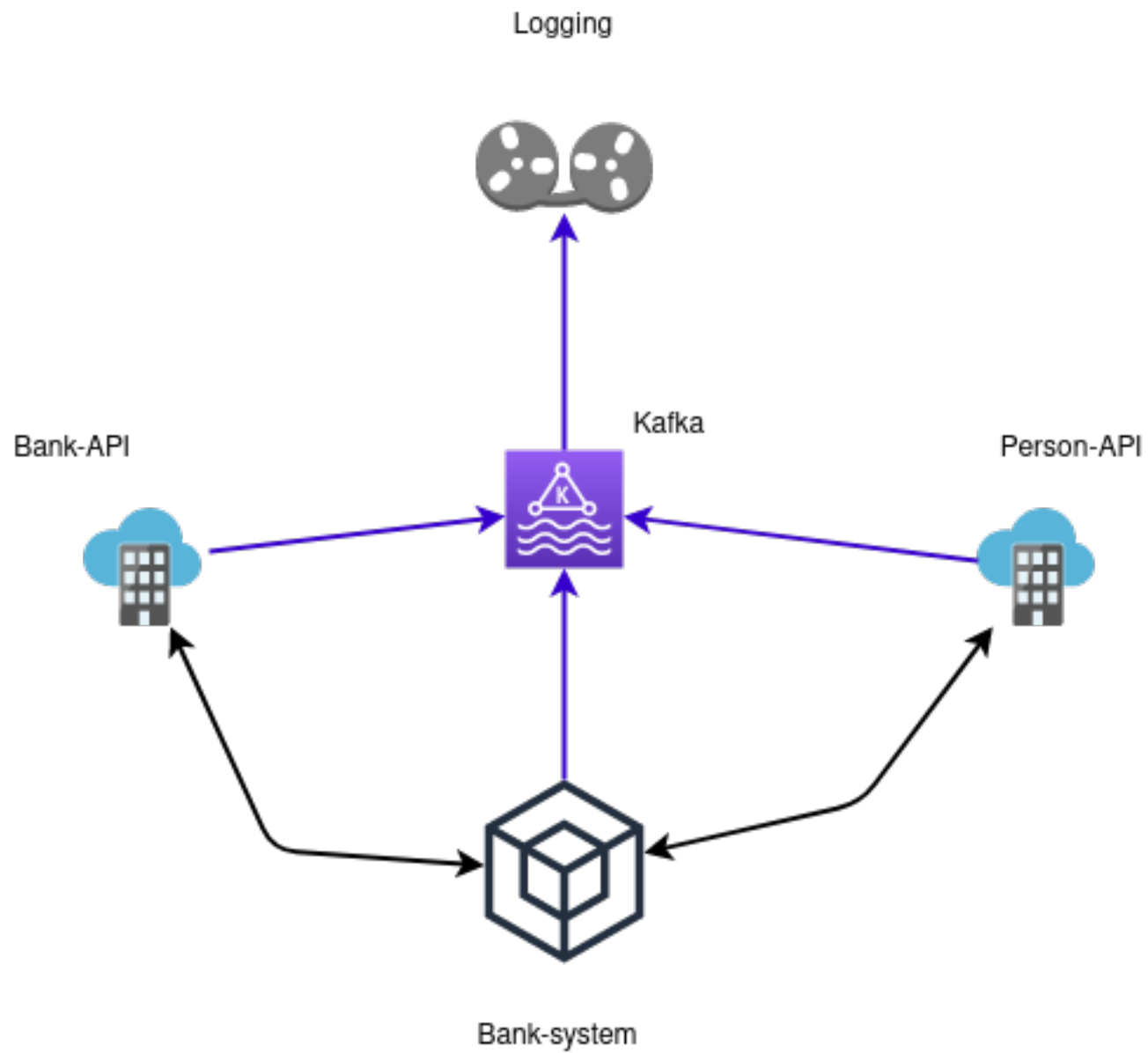
- Själva språket Java, dess syntax och standardbibliotek.
- Innehåller kompilator och andra verktyg.
- Koden kompileras till bytekod som exekveras i en virtuell maskin (JVM)

## Java EE

- Bibliotek för att bygga enterprise system.
- Server för tjänster att köra på.
- De delar som behövs för att lösa vanliga problem vid ES utveckling.

# Projektet

- Exakta instruktioner finns på kurshemsidan. Kom ihåg Webreg.
- Ni laddar ner ett ”kodskelett” från gitlab. Detta är ett antal projekt som innehåller de delarna ni behöver för att utveckla *data*, *logic* och *web*. Vi kommer i följande föreläsningar titta på exakt hur dessa utvecklas.



HTTP

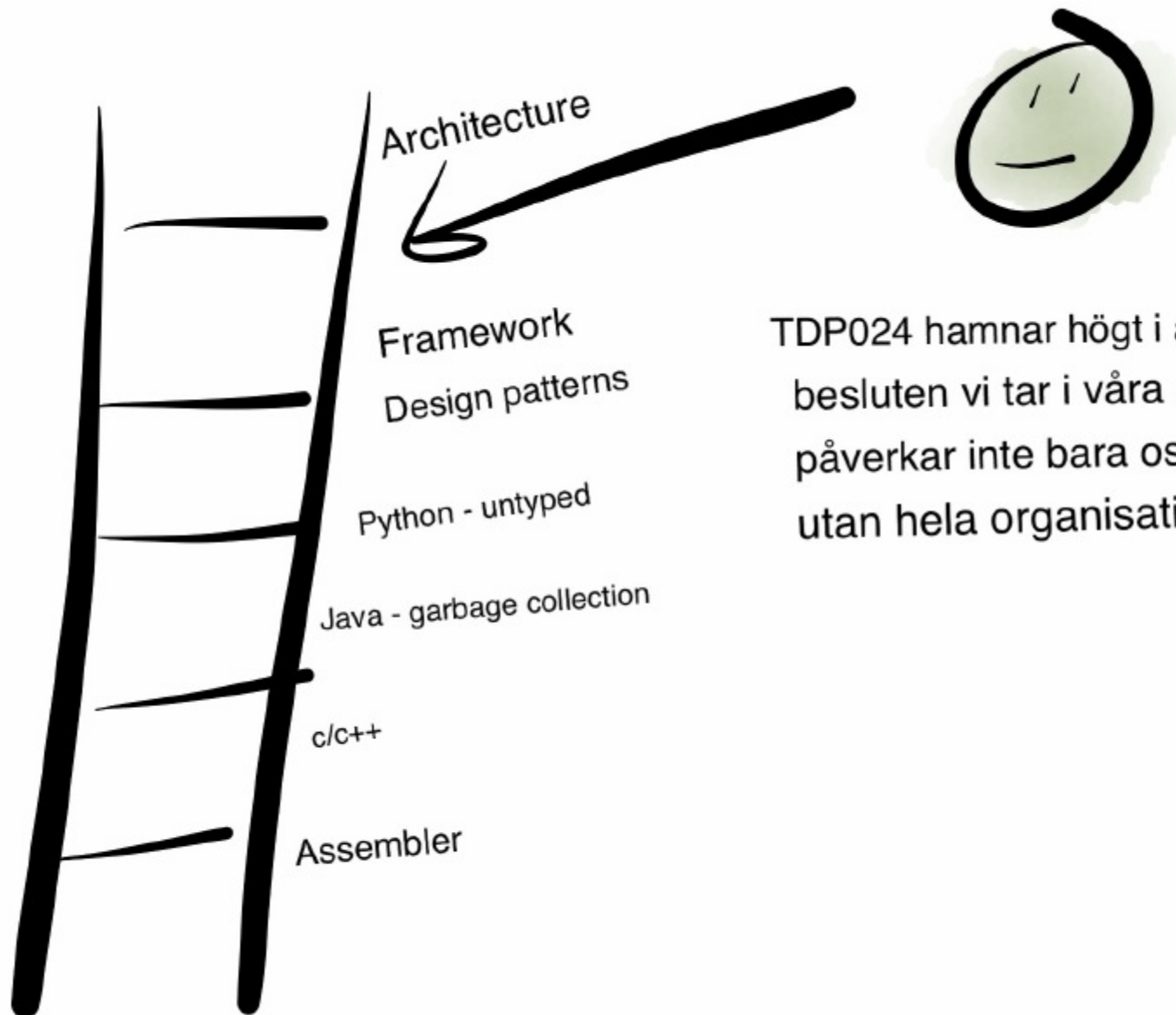


KAFKA

## Betyg 4 och 5

- Betyg 4 – Utöka ditt projekt felhantering (bara java koden)
- Betyg 5 - Betyg 4 + En avancerad del av Laboration 1 eller 2
- Redovisning sker genom muntlig presentation i laborationssalen till er assistent samt manuell genomgång av er kod av kursledning (kolla hemsidan för mer detaljer).

# Abstraktion



TDP024 hamnar högt i abstraktionen, besluten vi tar i våra scenarion påverkar inte bara oss själva utan hela organisationen.



# To get there: To reach but not overreach


- Code conventions - design principles - high-level structures


# Code conventions


```
def loadSchemas():  
    schemaTuples, badSchemaFiles = loadAllJsonObjects("schemas")  
    schemas = {}  
    for path, fileName, o in schemaTuples:  
        schemas[fileName[:5]] = o  
    return schemas, badSchemaFiles
```


Syntax and style


```
def load_schemas():  
    schema_tuples, bad_schema_files = load_all_json_objects("schemas")  
    schemas = {}  
    for path, file_name, o in schema_tuples:  
        schemas[file_name[:5]] = o  
    return schemas, bad_schema_files
```

 AbstractOption is a raw type. References to generic type AbstractOption<T> should be parameterized

 AbstractOption is a raw type. References to generic type AbstractOption<T> should be parameterized

 ArrayList is a raw type. References to generic type ArrayList<E> should be parameterized

 ArrayList is a raw type. References to generic type ArrayList<E> should be parameterized

 ArrayObjectQueue is a raw type. References to generic type ArrayObjectQueue<E> should be parameterized

Compiler warnings

```
int[] a = new int[] {1,2,3};  
int sum=0;  
for (int i : a) {  
    sum+=a[i];  
}
```

```
Arrays.stream(a).sum();
```

Language constructs

# Design principles

```

/** Constructor of the Cell class.
 * It is the only way to set the name of the cell.
 * @param name the name of the cell
 */
public Cell(String name) {
    this.name = name;
    formula = "";
    observers=new LinkedList<CellObserver>();
    anal = Analyzer.getInstance();
    state = EquationState.getInstance();
    astN = new ASTValue(0.0);
}

```

Coupling

- ▼  **Enumerable<T>**
  - ▣ iterable : Iterable<T>
  -  Enumerable(Iterable<T>)
  -  collect(T, ICollector<T>) : T
  -  forEach(IAction<T>) : void
  -  iterator() : Iterator<T>
  - ▶ ●  select(Selector<T, U>) <U> : IEnumerable<U>
  - ▶ ●  skip(int) : IEnumerable<T>
  - ▶ ●  skipUntil(IPredicate<T>) : IEnumerable<T>
  - ▶ ●  take(int) : IEnumerable<T>
  - ▶ ●  takeUntil(IPredicate<T>) : IEnumerable<T>
  - ▶ ●  where(IPredicate<T>) : IEnumerable<T>

Cohesion

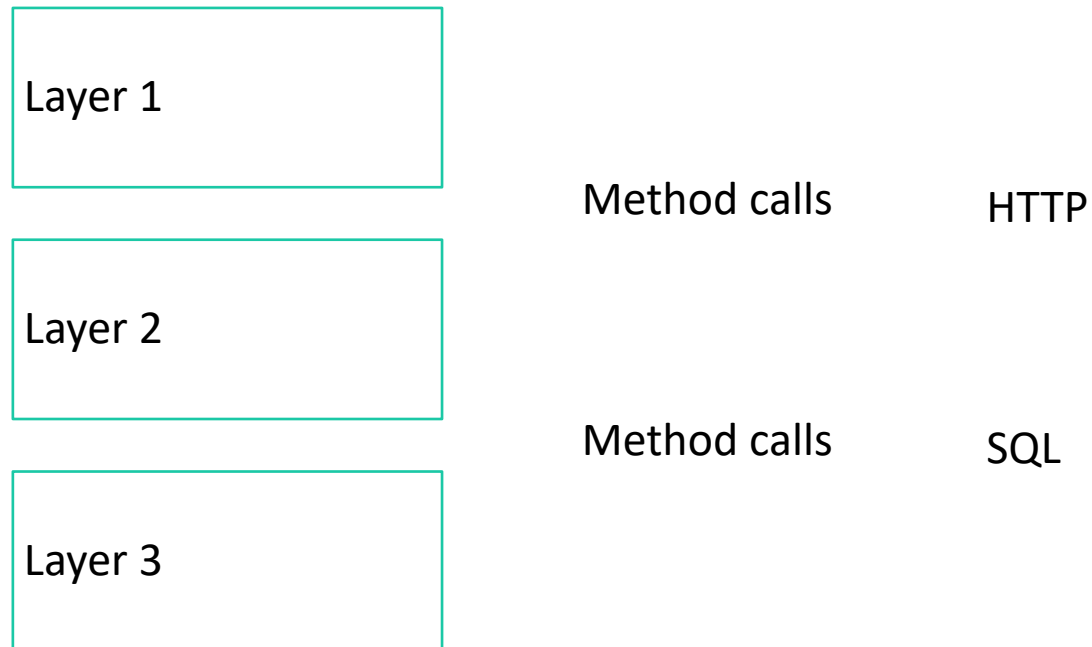
# Software architecture

- *”The set of structures needed to reason about the system, which comprises software elements, relations among them, and properties of both”* - Documenting Software Architectures: Views and Beyond (2nd Edition), Clements et al, Addison-Wesley, 2010
- *”the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution”* - ANSI/IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems
- *”That peculiar consistency and homogeneity of Design that would give rise to ease of use and reuse, which is wished for by the Developer who calls himself ”The Software Architect.””* - John Carter (Software Technologist, Tait Electronics, Christchurch, New Zealand)

# Software architectures

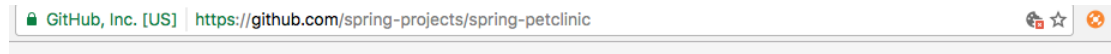
- Flexibility - estimated cost of making changes to individual parts
- Testability - estimated cost of devising tests for parts or the whole
- Scalability/Performance - estimated cost of changing the system to accommodate higher loads
- Cost of development - estimated cost of developing the common core
- Cost of deployment - estimated cost of running the system

# Layered architecture





# Spring PetClinic



## Spring PetClinic Sample Application build passing

### Understanding the Spring Petclinic application with a few diagrams

[See the presentation here](#)

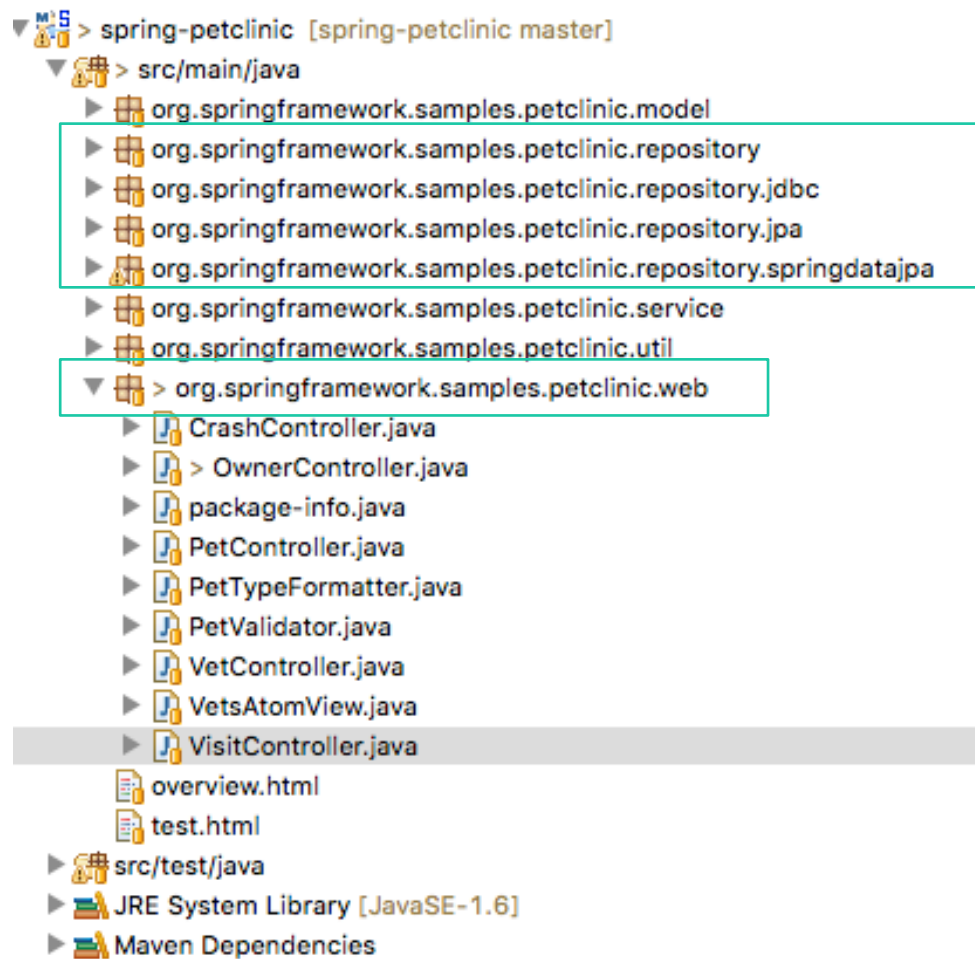
### Running petclinic locally

```
git clone https://github.com/spring-projects/spring-petclinic.git
cd spring-petclinic
./mvnw spring-boot:run
```

You can then access petclinic here: <http://localhost:8080/>

The screenshot shows the Spring PetClinic application interface. At the top, there is a navigation bar with the Spring logo and menu items: HOME, FIND OWNERS, VETERINARIANS, and ERROR. Below the navigation bar, the page title is "Owners". A table displays a list of pet owners with columns for Name, Address, City, Telephone, and Pets. The table contains 10 rows of data. At the bottom of the page, there is a footer with the Spring logo and the text "© 2013".

Name	Address	City	Telephone	Pets
Jeff Black	1450 Oak Blvd.	Monona	6085555387	Lucky
Jean Coleman	105 N. Lake St.	Monona	6085552654	Max Samantha
Betty Davis	638 Cardinal Ave.	Sun Prairie	6085551749	Basil
Harold Davis	563 Friendly St.	Windsor	6085553198	Iggy
María Escobito	345 Maple St.	Madison	6085557683	Mulligan
Carlos Estaban	2335 Independence La.	Waunakee	6085555487	Lucky Sly
George Franklin	110 W. Liberty St.	Madison	6085551023	Leo
Peter McFavish	2387 S. Fair Way	Madison	6085552765	George
Eduardo Rodriguez	2693 Commerce St.	McFarland	6085558763	Jewel Rosy
David Schroeder	2749 Blackhawk Trail	Madison	6085559435	Freddy



```

public class VisitController {

    private final ClinicService clinicService;

    public VisitController(ClinicService clinicService) {}

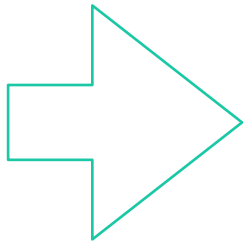
    public void setAllowedFields(WebDataBinder dataBinder) {}

    * Called before each and every @RequestMapping annotated method.
    @ModelAttribute("visit")
    public Visit loadPetWithVisit(@PathVariable("petId") int petId) {
        Pet pet = this.clinicService.findPetById(petId);
        Visit visit = new Visit();
        pet.addVisit(visit);
        return visit;
    }

    // Spring MVC calls method loadPetWithVisit(...) before initNewVisitForm is called
    @RequestMapping(value = "/owners/*/pets/{petId}/visits/new", method = RequestMethod.GET)
    public String initNewVisitForm(@PathVariable("petId") int petId, Map<String, Object> model) {
        return "pets/createOrUpdateVisitForm";
    }

    // Spring MVC calls method loadPetWithVisit(...) before processNewVisitForm is called
    @RequestMapping(value = "/owners/{ownerId}/pets/{petId}/visits/new", method = RequestMethod.POST)
    public String processNewVisitForm(@Valid Visit visit, BindingResult result) {
        if (result.hasErrors()) {
            return "pets/createOrUpdateVisitForm";
        } else {
            this.clinicService.saveVisit(visit);
            return "redirect:/owners/{ownerId}";
        }
    }
}

```

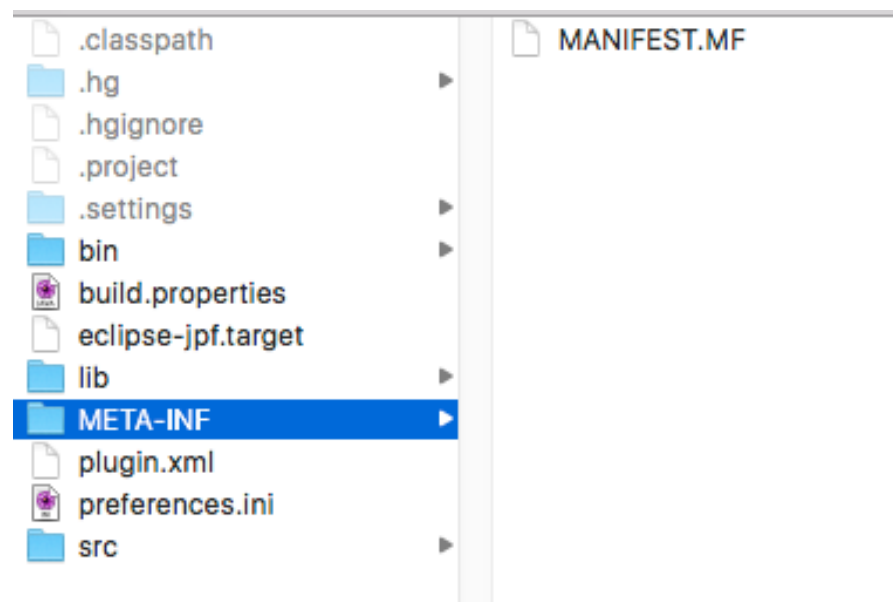


# Analysis

- Testability?
- Scalability?
- Flexibility?
- Ease of development?
- Ease of deployment?

# Microkernel architecture

- Eclipse - Equinox - OSGi
- VS Code



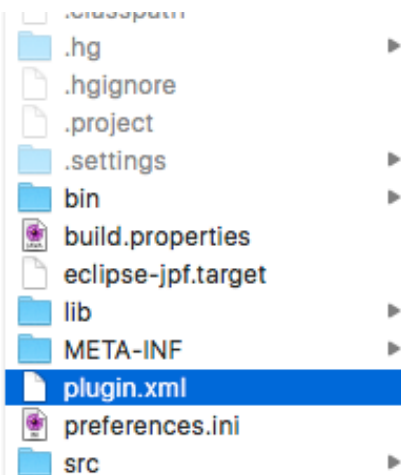
# Manifest

54

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: Eclipse-jpf
Bundle-SymbolicName: eclipse-jpf;singleton:=true
Bundle-Version: 1.0.0.qualifier
Bundle-ClassPath: lib/antlr-runtime-3.4.jar,
    lib/ST-4.0.4.jar,
    lib/jpf-template.jar,
    lib/RunJPF.jar,
```

```
Bundle-Activator: eclipse_jpf.Activator
Export-Package: eclipse_jpf,
    gov.nasa.runjpf,
    gov.nasa.runjpf.options,
    gov.nasa.runjpf.util,
    gov.nasa.runjpf.wizard
Require-Bundle: org.eclipse.ui,
    org.eclipse.core.runtime,
    org.eclipse.ui,
    org.eclipse.core.runtime,
    org.eclipse.ui,
    org.eclipse.core.runtime
Bundle-RequiredExecutionEnvironment: JavaSE-1.8
Bundle-ActivationPolicy: lazy
Import-Package: org.eclipse.core.filesystem,
    org.eclipse.core.resources,
    org.eclipse.core.runtime;version="3.4.0",
    org.eclipse.core.runtime.jobs,
    org.eclipse.core.runtime.preferences;version="3.3.0",
    org.eclipse.jdt.core,
    org.eclipse.jface.action,
    org.eclipse.jface.dialogs,
    org.eclipse.jface.preference,
    org.eclipse.jface.text,
    org.eclipse.jface.viewers,
    org.eclipse.jface.wizard,
    org.eclipse.osgi.util;version="1.1.0",
    org.eclipse.swt,
    org.eclipse.swt.events,
    org.eclipse.swt.graphics,
    org.eclipse.swt.layout,
    org.eclipse.swt.widgets,
    org.eclipse.ui,
    org.eclipse.ui.console,
    org.eclipse.ui.dialogs,
    org.eclipse.ui.ide,
```

# Extension points



```
<?xml version="1.0" encoding="UTF-8"?>
<?eclipse version="3.4"?>
<plugin>
  <extension
    point="org.eclipse.core.contenttype.contentTypes">
    <content-type
      base-type="org.eclipse.core.runtime.properties"
      file-extensions="jpf"
      id="com.javapathfinder.vjp.configContentType"
      name="JPF Config File"
      priority="normal">
    </content-type>
  </extension>
  <extension
    point="org.eclipse.ui.popupMenus">
    <objectContribution
      adaptable="false"
      id="runjpf.jpfConfigLaunch"
      nameFilter="*.jpf"
      objectClass="org.eclipse.core.resources.IFile">
      <action
        class="gov.nasa.runjpf.VerifyActionDelegate"
        enablesFor="1"
        id="RunJPF.verifyaction"
        label="Verify..."
        menubarPath="additions"
        tooltip="Execute this property file as a JPF configuration">
      </action>
    </objectContribution>
  </extension>
  <extension
    point="org.eclipse.ui.preferencePages">
    <page
      class="gov.nasa.runjpf.options.Preferences"
      id="eclipse-jpf.preferences"
      name="JPF Preferences">
    </page>
  </extension>
  <extension
    point="org.eclipse.core.runtime.preferences">
    <initializer class="gov.nasa.runjpf.options.DefaultPreferences" />
  </extension>
  <extension
    point="org.eclipse.ui.newWizards">
    <category
      id="eclipse-jpf.category.wizards">
    </category>
  </extension>
</plugin>
```

# Lifecycle

56

```
public class EclipseJPF extends AbstractUIPlugin {

    /**
     * The plug-in ID
     */
    public static final String PLUGIN_ID = "RunJPF";

    // The shared instance
    private static EclipseJPF plugin;

    public EclipseJPF() {
    }

    /**
     * (non-Javadoc)
     * @see org.eclipse.ui.plugin.AbstractUIPlugin#start(org.osgi.framework.BundleContext)
     */
    public void start(BundleContext context) throws Exception {
        super.start(context);
        plugin = this;
    }

    /**
     * (non-Javadoc)
     * @see org.eclipse.ui.plugin.AbstractUIPlugin#stop(org.osgi.framework.BundleContext)
     */
    public void stop(BundleContext context) throws Exception {
        plugin = null;
        super.stop(context);
    }
}
```



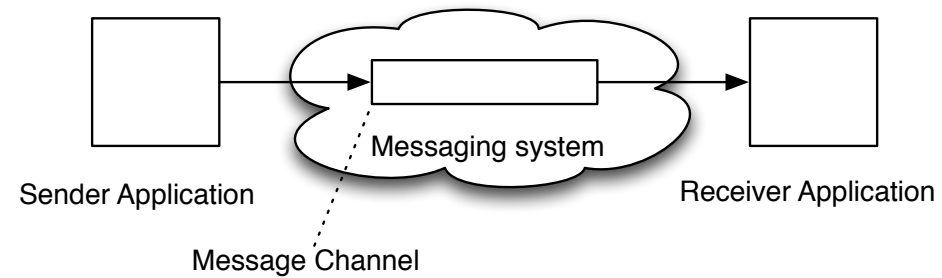
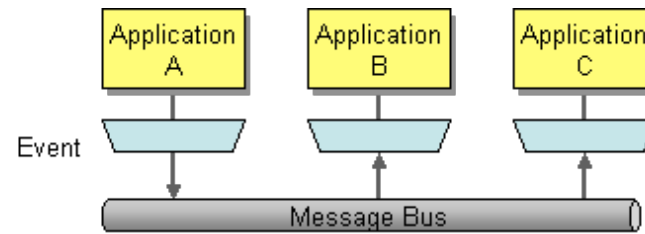
# Plugin

```
public class DefaultPreferences extends AbstractPreferenceInitializer {  
  
    @Override  
    public void initializeDefaultPreferences(){  
        IPreferenceStore store = EclipseJPF.getDefault().getPreferenceStore();  
        store.setDefault(EclipseJPFLauncher.SITE_PROPERTIES_PATH,  
EclipseJPFLauncher.DEFAULT_SITE_PROPERTIES_PATH);  
        store.setDefault(EclipseJPFLauncher.PORT, EclipseJPFLauncher.DEFAULT_PORT);  
    }  
}
```

# Analysis

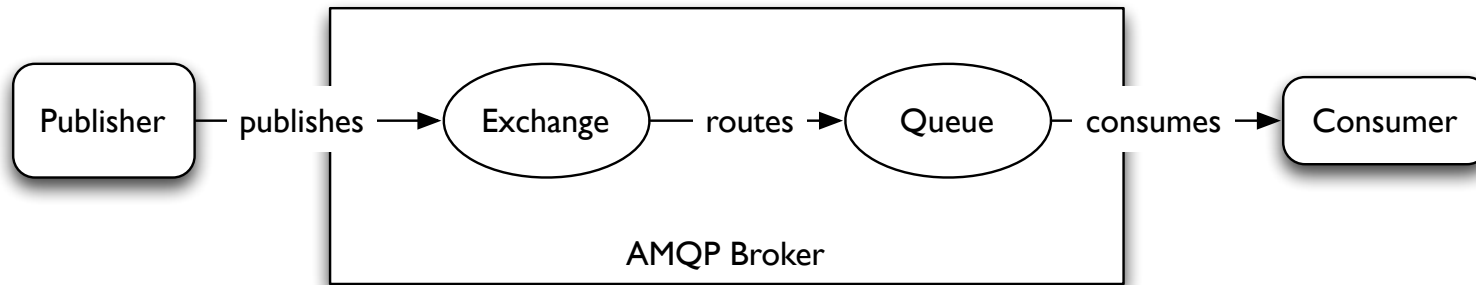
- Testability?
- Scalability?
- Flexibility?
- Ease of development?
- Ease of deployment?

# Message bus architectures

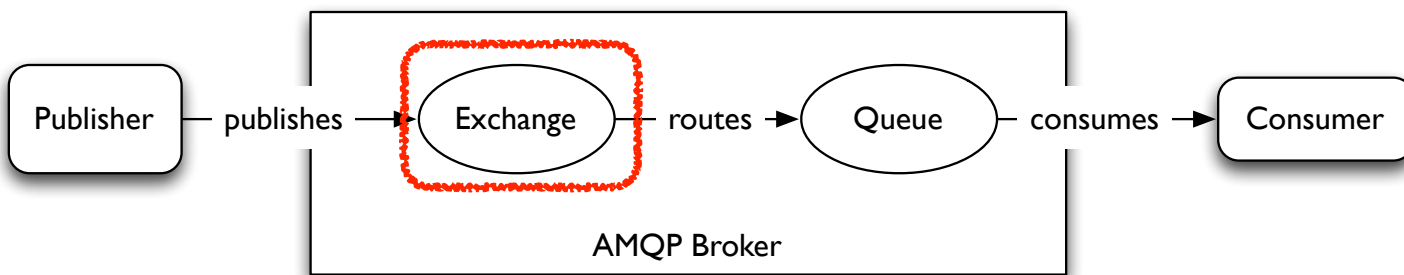




# AMQP: Advanced Message Queueing Protocol



<https://www.rabbitmq.com/tutorials/amqp-concepts.html>



# Analysis

- Testability?
- Scalability?
- Flexibility?
- Ease of development?
- Ease of deployment?

# Summering

- HSV examensmål och koppling till kursens uppbyggnad
- Vi definierade: Multi-Tier, SOA , Enterprise Systems, Software Architecture
- Vi tittade på SOA principer: Contract, Abstraction, Autonomy och Stateless
- Vi tittade på hur vi kan utvärdera Mjukvaruarkitekturer



# Linköpings universitet

expanding reality

[www.liu.se](http://www.liu.se)