<table>
<thead>
<tr>
<th>framework</th>
<th>ejb</th>
<th>session management &amp; authentification</th>
</tr>
</thead>
<tbody>
<tr>
<td>architecture</td>
<td>exceptions</td>
<td>versioning</td>
</tr>
<tr>
<td>services</td>
<td>logging</td>
<td>bug management</td>
</tr>
</tbody>
</table>
frameworks

Design Principles

lightweight
as simple as possible
based on proven packages
base level functionality only
extendable, loadable services
vendor independent
cross application server portability
overall deployment

- customization package
- interactive format platform
- service framework
- java 2 EE
- service adapters
- operating system
- application server
- legacy systems
integration / example

loadable services for easy integration into existing systems
basic architecture

Java Virtual Machine / Server

services=scheduler,logging
scheduler.class=at.sysis.framework.service.scheduler.SchedulerService
logging.class=at.sysis.framework.service.logging.JRunLoggingService
logging.test=This is a very simple logger

services.xml
framework services

for external responsibilities
to be used as adapters or bridges
easy to be implemented
to customize an application by changing configuration
services are local to a web application, publishing itself
ServiceContext is the registry for services
usually has an interface, can be initialized and destroyed
one or more implementations, implement interface Service
startup servlet as starting point of the application
3rd party libraries are hidden behind services
service lookup

locate the registry 1:
ServiceContext serviceContext =
    (ServiceContext)
    getServletContext().getAttribute(ServiceContext.ROOT_CONTEXT_NAME);

locate the registry 2:
    (when using custom authentication, not recommended, only one appl. per vm)
InitialContext context = new InitialContext();
Authentication authenticator = (Authentication)
    context.lookup("authentication");

locate a service:

IdGenerator idGenerator = (IdGenerator)
    serviceContext.findService("idgenerator")
EJB integration

some EJB generators
http://sourceforge.net/projects/ejbdoclet/
http://www.beust.com/cedric/ejbggen/

some restrictions imposed on EJB 1.1
no thread management, no thread sync
no read/write static fields
no java.IO
avoid listening to sockets, no accept on connections, no multicast
no native library access
...
EJB integration BEA Weblogic

simple example:

```java
Object home =
    initialContext.lookup("at.sysis.consumer.registration.ConsumerRegistrarHome");

consumerRegistrarHome =
    (ConsumerRegistrarHome)PortableRemoteObject.narrow(home,
             ConsumerRegistrarHome.class);

ConsumerRegistrar con = consumerRegistrarHome.create();

registerEntry =
    consumerRegistrarAdmin.getRegisterEntryForUserid(userId);
```
exception handling

use Java exceptions only for errors that are not expected

expected errors should be part of the Use Case description

ServiceException is a container for other exceptions

XXXServiceError derived from AbstractServiceError

sample

throw new

ServiceException(InternalServiceError.CLASS_NOTFOUND, e);
Logging / log4j

Requirements

- ease of use
- flexible output formatting
- different channels
- no static dependency
Logging / log4j

Priorities

from left to right, including all other priorities
logging

category is fully qualified name of class which logs something.
priorities have a hierarchy. DEBUG level shows all priorities
failed assertion is logged with priority ERROR.
appenders produce the actual logging output
(FileAppender, AsyncAppender, SMTPAppender, etc.)
logging example

non-static:

```java
Category cat = logger.getCategory(MyClass.class.getName());
cat.debug("Hallo World");
```

static (esp. for debugging):

```java
static Category CAT =
    StaticLogger.getCategory(MyClass.class.getName());
```
session management

sysis originally implemented own session management

currently propose use of Servlet API Session Management

enables integration with 3rd Party Components

enables the use of Servlet API Authorization and Authentication

SessionBindingEvents useful for notifications on timeout

notification for bound and unbound from the session

avoid binding a user directly to a unique session!
authentification and authorization

Servlet API: getRemoteUser, isUserInRole

- Basic Authentication for Administration Tools
- Form based Authentication for Applications
- J Run AuthenticationInterface for custom authentication
  (Tomcat has something similar)
versioning / meta inf

prepare a manifest file and compile it into jar

```
jar cvfm %archivename%\abs_pfad\austropolis.mf *
```

read version from manifest of jar

```
import java.util.jar.*;

try {
    JarInputStream inputStream =
        new JarInputStream(new FileInputStream("austropolis.jar"));
    manifest = inputStream.getManifest();
} catch (Exception e) { }

Attributes att = manifest.getMainAttributes();
implversion = att.getValue("Implementation-Version");
```
versioning / meta inf

$Name: $ will be filled in automatically by CVS when you checkout the file.  
E.g. it will become $Name: framework-2-0-3 $

attributes

Name: at/sysis/netlife/flirtboat2
Specification-Title: Flirtboat 2 Classes
Specification-Version: 2.1
Specification-Vendor: sysis ag
Implementation-Title: at/sysis/netlife/flirtboat2
Implementation-Version: $Name: $  
Implementation-Vendor: sysis ag
development folder

<project name>

- distribution
- lib
- docs
- sql
- media
- src
- web-app
- bin
- production
  - web-inf
    - jsp
    - classes
    - lib
  - web
    - media
  - wap
    - media
error management with bugzilla
development method

domain engineering: diagram shows packages instead of milestones