Tutorial: Google App Engine

SYSTEMATIC THOUGHT LEADERSHIP FOR INNOVATIVE BUSINESS

Benjamin Schmeling, SAP Research/Technical University Darmstadt

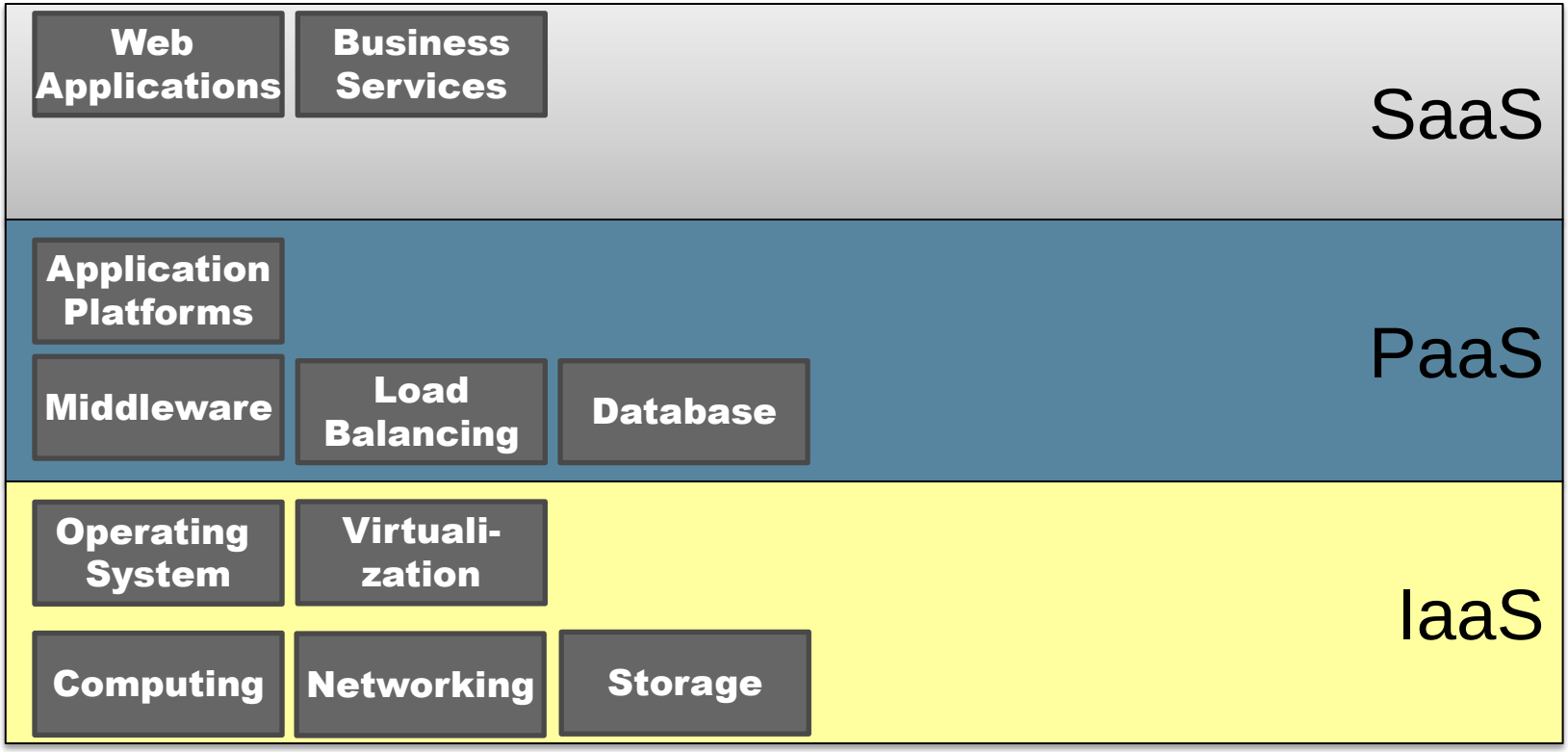
4th DAAD Summer School on Current Trends in Distributed Systems

Sousse. Tunisia, September 6th 2012

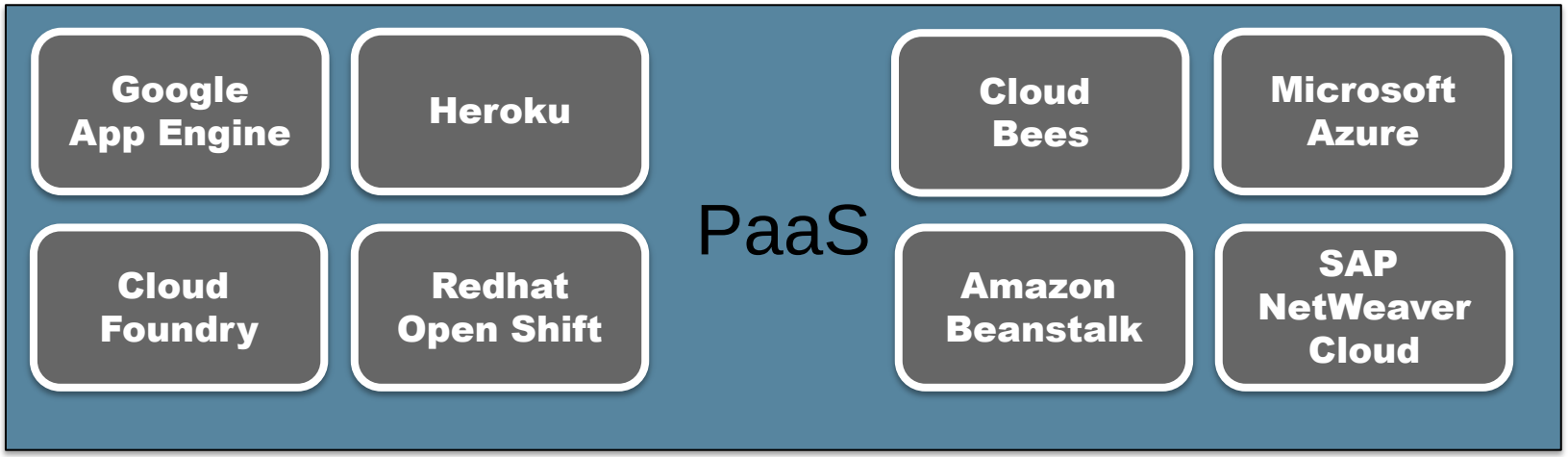
Outline

- Introduction to Google App Engine
- Creating a new project
- Persistence using NoSQL Data Store
  - Datastore API
  - JPA
- Creating a Web Service
- (Web User Interface with JSF/Richfaces)
- (Management of deployed Applications)

Introduction
Platform as a Service



Introduction
Platform as a Service Providers



- Providers offer support for different…
  - Programming Languages (Java, C#, Python, PHP, Groovy, Ruby, …)
  - Databases (MySQL, PostgreSQL, Cassandra, MongoDB, HSQL, MS SQL, ..)
  - Platform Services (Authentication, Email, Payment/Billing, …)
  - Pricing (Free, Bandwidth, Storage per Time, CPU Time, Email, …)

Introduction
Why Google App Engine?

- Web Applications on Google‘s infrastructure
  - Platform as a Service (PaaS) offering
  - Scalable in terms of data and computing power
  - Different programming languages supported:
    - Java
    - Python
    - Go
  - Pure pay-per-use model based on bandwidth and storage
    - 1GB storage 5 million page views for free

Introduction
Features & Limitations

- JRE
  - Java 5 or 6 supported
  - Java Data Objects (JDO) and Java Persistence API (JPA) can be used for persistence
  - Java Mail API can be used
- Data Store
  - Distributed NoSQL data storage service
  - Transactional (ACID)
  - No Database Schema
- Every application runs in a reliable and secure sandbox
  - An application can only access other computers on the Internet through the provided URL fetch and email services
  - Only HTTP (or HTTPS) on standard ports is supported to access applications
  - Applications cannot write to the file system
  - Only files can be read which have been uploaded with the application code
  - For persistence App Engine datastore, memcache or other services have to be used
  - App must return response data within 60 seconds

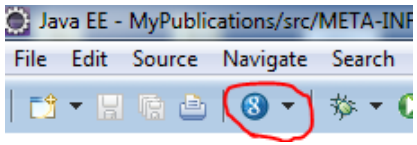
App Engine Services

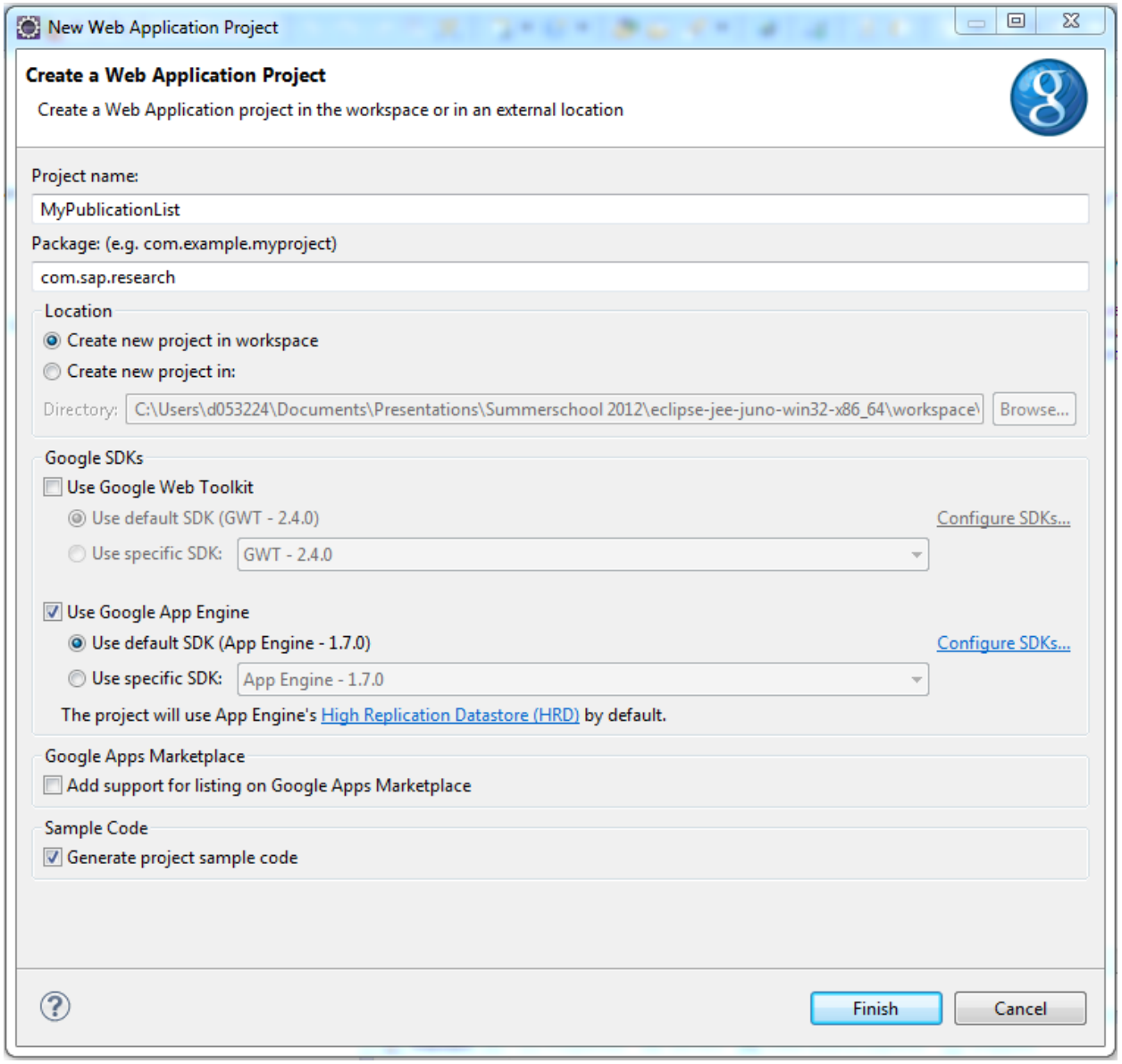
- Users Authentication based on Google Accounts
- Multitenancy
- Capabilities
- URL fetch to access resources on the Internet (e.g. web services)
- Mail
- App Identity
- LogService
- Memcache
- Image Manipulation such as resize, crop, rotate flip JPEG and PNG images
- Conversion
- OAuth
- Search
- Prospective Search
- XMPP

Getting started
IDE Installation

- Download Eclipse Juno JEE from http://www.eclipse.org/downloads/
- Add Update Site http://dl.google.com/eclipse/plugin/4.2
- Install features
  - Google Plugin for Eclipse
  - Google App Engine Java SDK
  - Google Web Toolkit SDK

Getting started
Creating a new Project





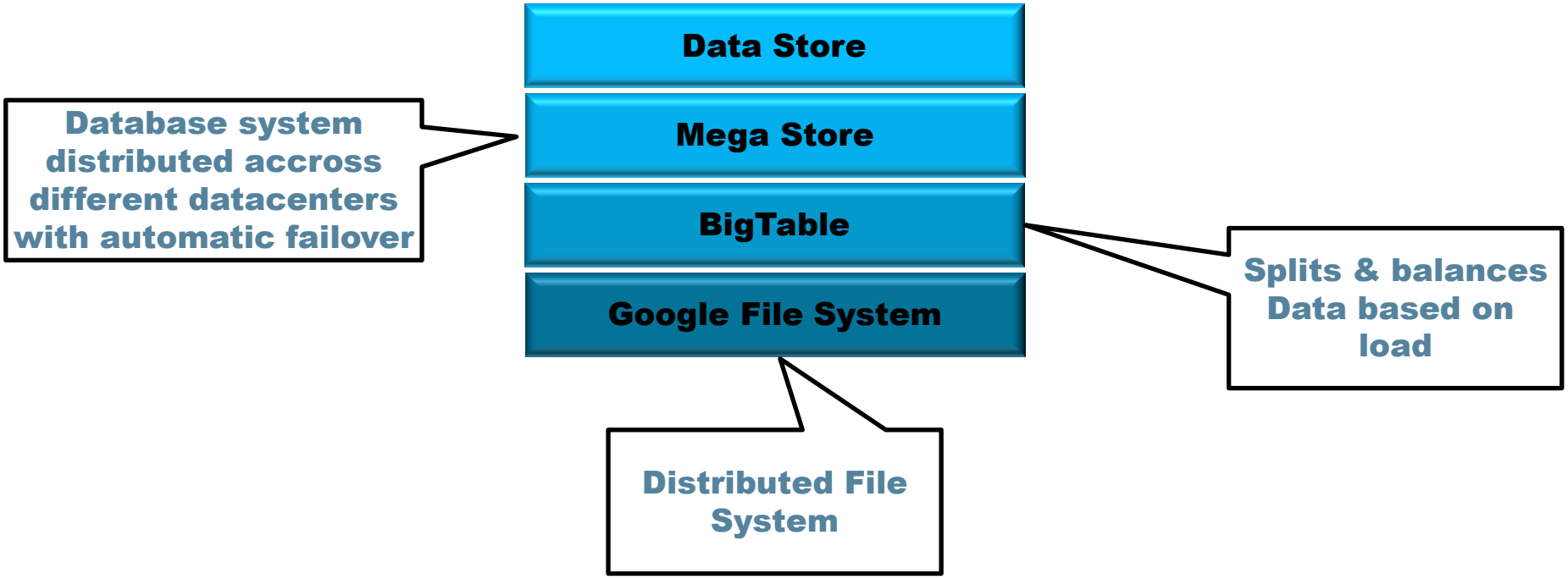
Persistence
Introduction

- There are three ways to store data in GAE
  - **App Engine Datastore**
    - NoSQL
    - *High Replication Datastore (HRD)*
  - Google Cloud SQL
    - Relational SQL Database based on MySQL
    - Charged Service (per package or per use)
  - Google Cloud Storage
    - Storage for large objects and files in buckets
    - Modification of objects not possible (overwrite required)
    - RESTful API

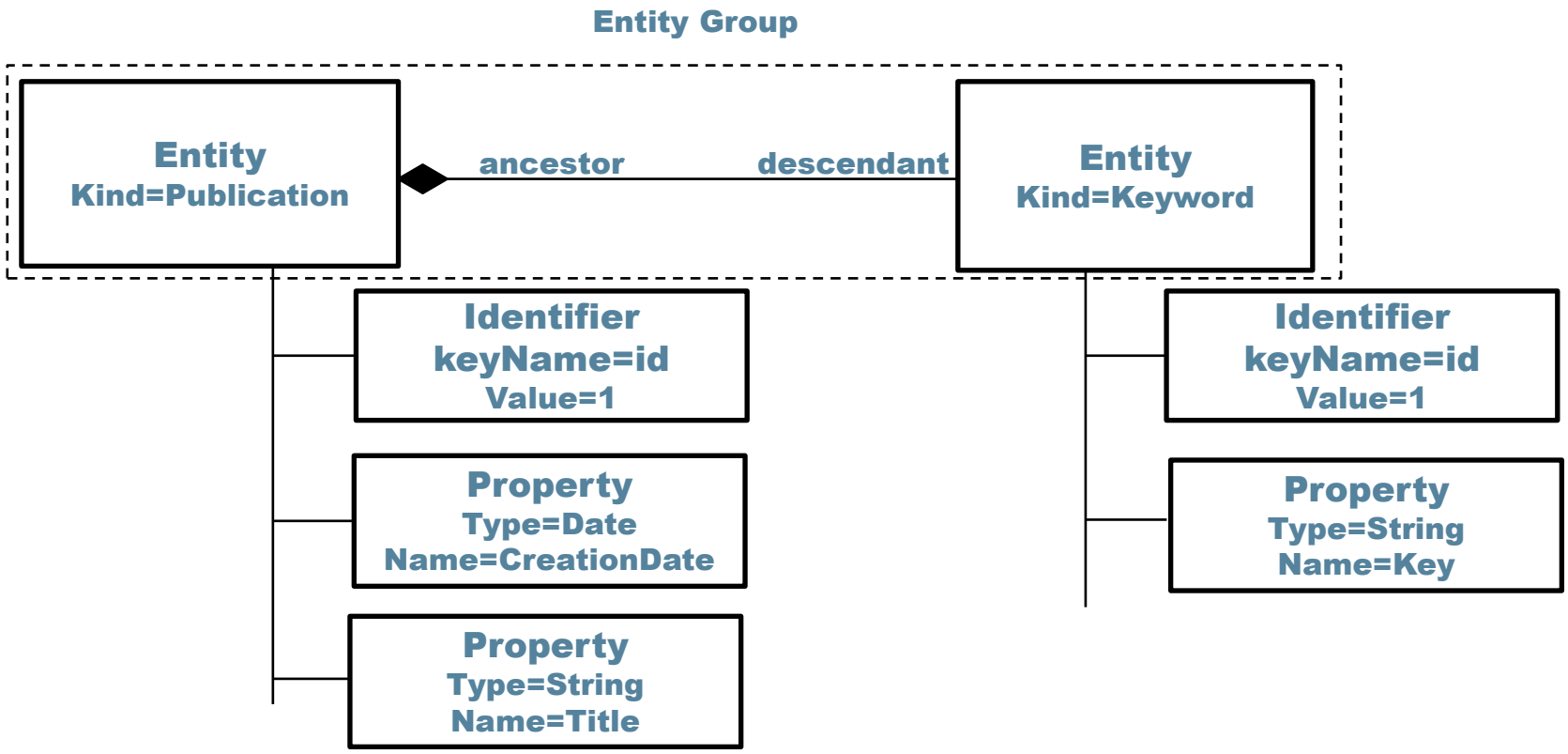
App Engine Datastore vs. Cloud SQL

Criterion	App Engine Datastore	Cloud SQL
Queries	(SQL)/MapReduce	Full SQL, Aggregations, Joins
Transactions	Entity Groups/5 Cross-Entity	ACID Transactions
Consistency	Eventual Consistency	Strict Consistency
Scalability	Accross Different Datacenters	Master/Slave (synchronous)
Management	Easy (No Schema)	SQL (Create, Alter …)
Schema	Can Be Enforced By App	Strictly Enforced

App Engine Datastore



Persistence
App Engine Datastore



Web Services

- Implementation based on
  - javax.xml.soap
  - javax.xml.bind.JAXB
- Skeleton and Stub Generation based on
  - javax.jws and wsgen/wsimport

JSF and AppEngine
Modified JSF Impl required

```
java.lang.NoClassDefFoundError: javax.naming.InitialContext is a restricted class.
    Please see the Google  App Engine developer's guide for more details at
    com.google.appengine.tools.development.agent.runtime.Runtime.reject(Runtime.jav
    a:51)

at
    com.sun.faces.config.WebConfiguration.processJndiEntries(WebConfiguration.java:
    685)

at com.sun.faces.config.WebConfiguration.<init>(WebConfiguration.java:134)

at com.sun.faces.config.WebConfiguration.getInstance(WebConfiguration.java:194)

at
    com.sun.faces.config.ConfigureListener.contextInitialized(ConfigureListener.jav
    a:163)

at org.mortbay.jetty.handler.ContextHandler.startContext(ContextHandler.java:548)
```

Setup Richfaces

- Add libs to WEB-INF/lib folder
  - richfaces-components-api.jar
  - richfaces-components-ui.jar
  - richfaces-core-api.jar
  - richfaces-core-impl.jar
  - cssparser.jar
  - sac.jar
  - guava.jar

- Richfaces Showcase: http://showcase.richfaces.org/richfaces/component-sample.jsf
- https://community.jboss.org/wiki/HowToUseRichFaces40WithGoogleAppEngine