

# Wahlzeit Design

**Prof. Dr. Dirk Riehle**

**Friedrich-Alexander University Erlangen-Nürnberg**

**ADAP A04**

Licensed under [CC BY 4.0 International](https://creativecommons.org/licenses/by/4.0/)

# Overall Architectural Style

- Servlet-based web application running on Google App Engine
- Persistence is managed by GAE using “Cloud Datastore”
- The architecture follows a typical model-view-separation style

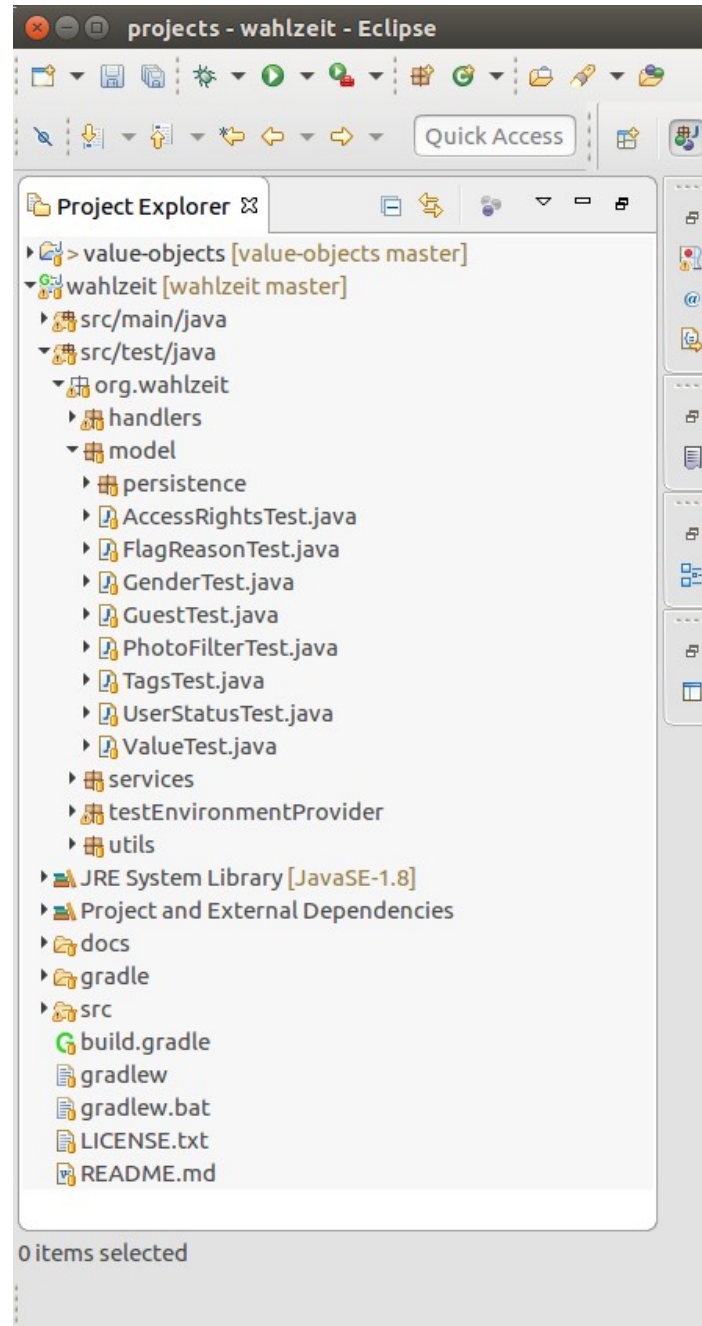
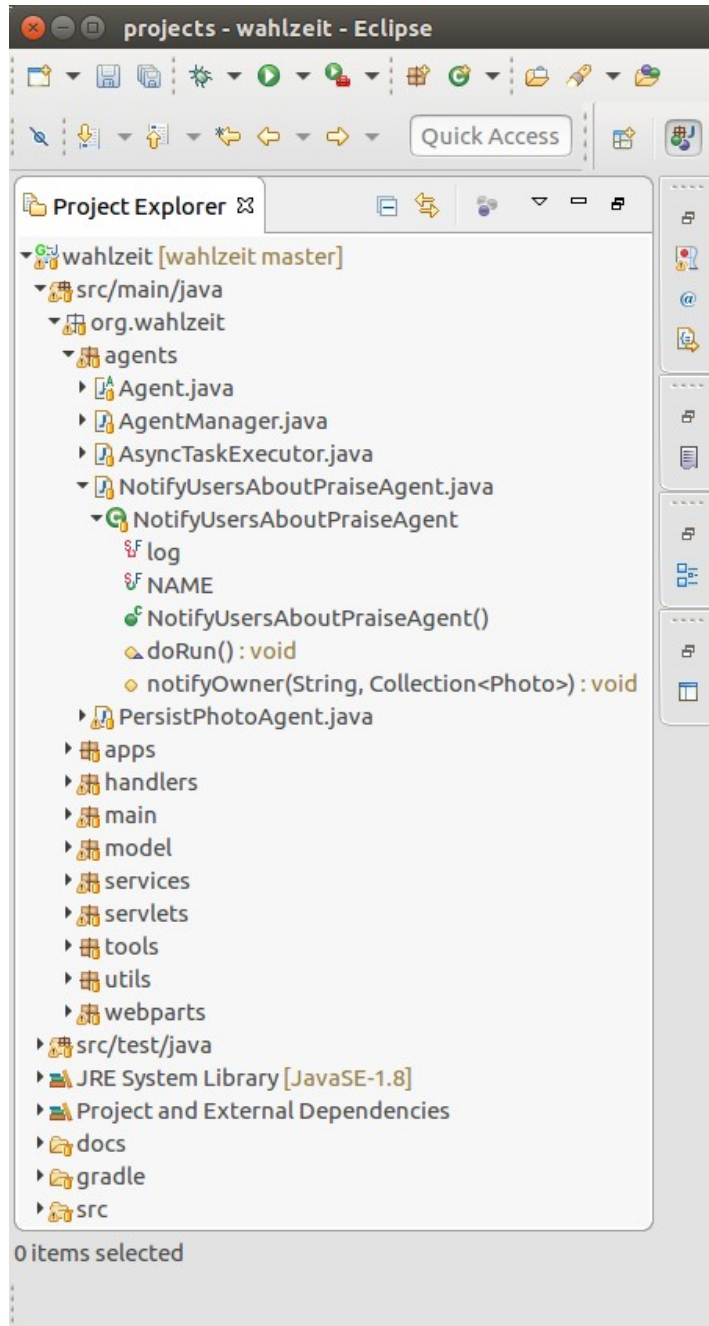
# “Physical” Architecture

- Client (browser)
- Application (application server)
- Database (database server)

# Application Architecture

- User interface (view)
- Domain model
- Database access

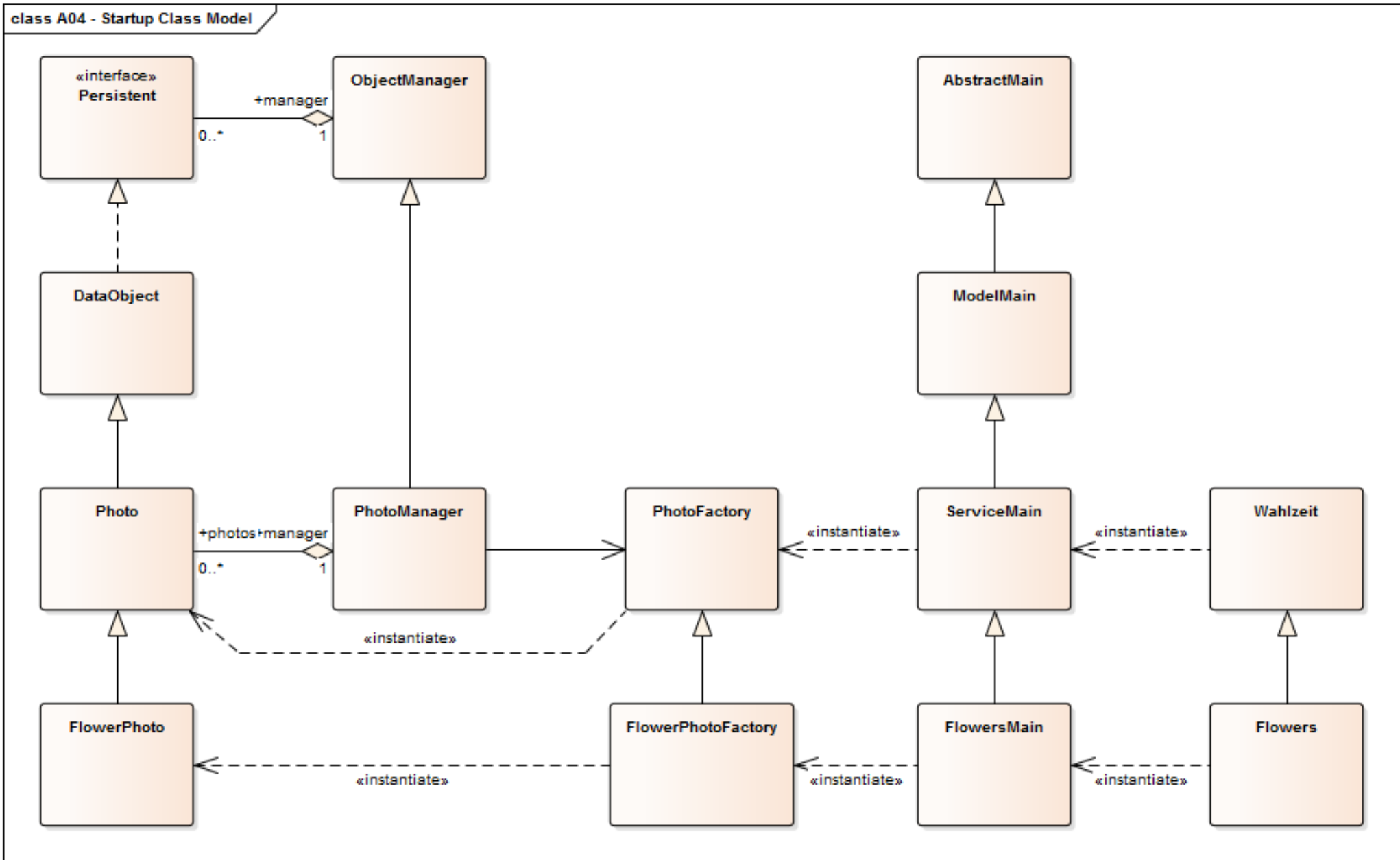
# Code Organization / Architecture



# Application Startup

- Wahlzeit (org.wahlzeit.apps)
  - A simple class hooked into the servlet container as listener
    - Listens to initialized and destroyed events
    - Delegates application startup and shutdown to Main class
- Main (org.wahlzeit.main)
  - A class hierarchy that handles system startup and shutdown
  - ServiceMain represents the main web service application
    - Starts-up all services; shuts them down upon request
  - ScriptMain provides a single faceless startup/shutdown cycle
    - Used to run scripts, for example, to migrate database schema

# Application Startup Class Model



# User Interaction

- Servlets (org.wahlzeit.servlets)
  - Default servlet handles static resource requests
  - Main Wahlzeit servlet handles all dynamic content requests
    - Has been configured with handlers per request type
    - Dispatches request to proper handler
- Handler (org.wahlzeit.handlers)
  - One handler for each page type of the application
    - Receives request information, performs handler-specific action
    - Knows how to call the domain model



# Web Page Generation

- Handlers (org.wahlzeit.handlers)
  - One handler for each page type of the application
    - Is configured with a web page (HTML) template
    - Requests templates and builds the web page
    - Utilizes a UserSession for a user's session state
- WebParts and Templates (org.wahlzeit.webparts)
  - A web-part is an object wrapping an HTML component
    - Is generated from a web-part template
  - All web-part templates are loaded from the file system
  - The WebPartTemplateService provides those templates

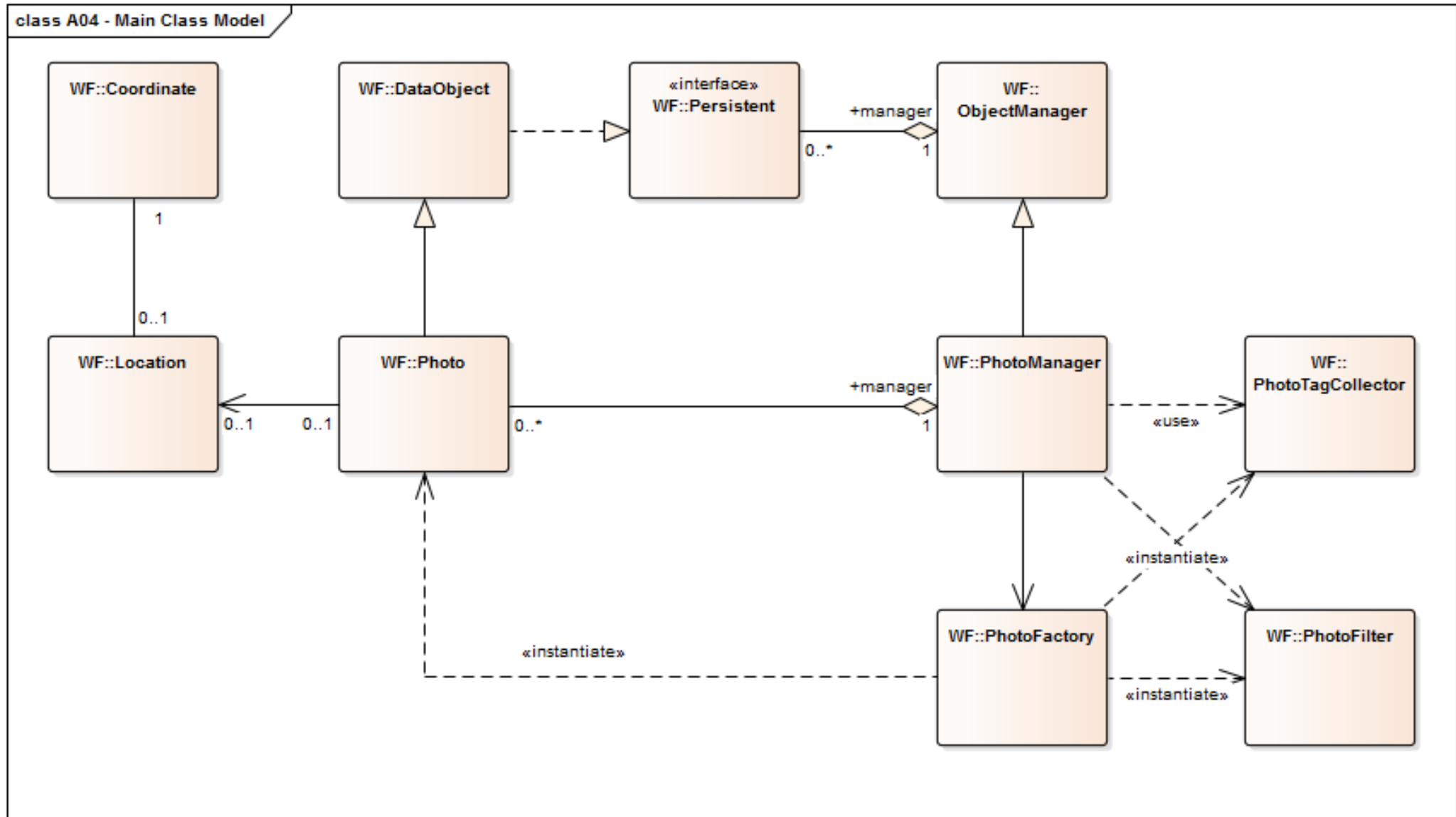
# Access Rights

- Client (org.wahlzeit.model)
  - Each visitor is represented by an object from the Client class hierarchy
    - Has appropriate AccessRights associated with it
- Handler (org.wahlzeit.handlers)
  - Each handler knows about the current user by way of the UserSession
    - Each access to a web page or a web-part is checked against user rights

# Domain Model

- Domain abstractions (org.wahlzeit.model)
  - Package contains all domain abstractions
    - Client hierarchy for visitors, logged-in users, moderators, administrators
    - Photo class hierarchy for photos
    - PhotoCase for cases filed against photos (copyright violation etc)
    - Various value object classes, e.g. language, status, gender
- Manager (org.wahlzeit.model)
  - An ObjectManager manages a particular domain abstraction (hierarchy)
    - UserManager manages the User class hierarchy objects, etc.
  - The object manager also hides the database access layer

# (Part of) Domain Class Model



# Object Persistence

- Persistent (org.wahlzeit.services)
  - Defines interface for reading and writing objects from storage
  - Provides dirty flag to indicate that object needs to be saved
- ObjectManager (org.wahlzeit.services)
  - Manages DataObject (implements Persistent) instances
    - One instance (of subclass) for each type of DataObject (subclass)
  - Encapsulates storage layer access

# System Configuration

- SysConfig (org.wahlzeit.services)
  - A key/value store with basic system configuration data
    - Provides key directories and URLs
- ModelConfig (org.wahlzeit.model)
  - A key/value store with model configuration information
    - Loaded from properties file
    - One for each language (currently only 'en' and 'de')
    - Provides language-specific text message formatting
- ConfigDir (org.wahlzeit.services)
  - Directory abstraction that provides custom extension mechanism

# Session Management

- Session (org.wahlzeit.services)
  - A session object maintains one database connection
  - For each active thread, one session object will be active
  - Session objects get allocated when thread becomes active
- SysSession (org.wahlzeit.services)
  - A session associated with faceless system threads
- UserSession (org.wahlzeit.services)
  - A session associated with each currently active user
  - Maintains user's session state, e.g. form input data

# Information Logging

- LogBuilder (org.wahlzeit.services)
  - A simple logging class that utilizes GAE logging



# Email Sending

- EmailService (org.wahlzeit.services.mailing)
  - Provides interface for sending email, various implementations:
    - SmtplibEmailService actually sends email
    - LoggingEmailService wraps email service and logs before sending
    - MockEmailService drops any request; needed for testing
- EmailServiceManager (org.wahlzeit.services.mailing)
  - Provides global email service instance
  - Can be configured for different execution scenarios (testing vs production)

# System Agents

- Agent (org.wahlzeit.agents)
  - An agent is a thread performing some faceless system functionality
    - Runs in the background, sleeps most of the time
- AgentManager (org.wahlzeit.agents)
  - Manages available agents, wakes them up

# Photo Praise Notification

- PraisePhotoFormHandler (org.wahlzeit.handlers)
  - If a photo owner requested praise notification, enqueues praise with agent
- NotifyAboutPraiseAgent (org.wahlzeit.agents)
  - Agent enqueues praise requests received from handler
    - Sends out praise information in regular intervals

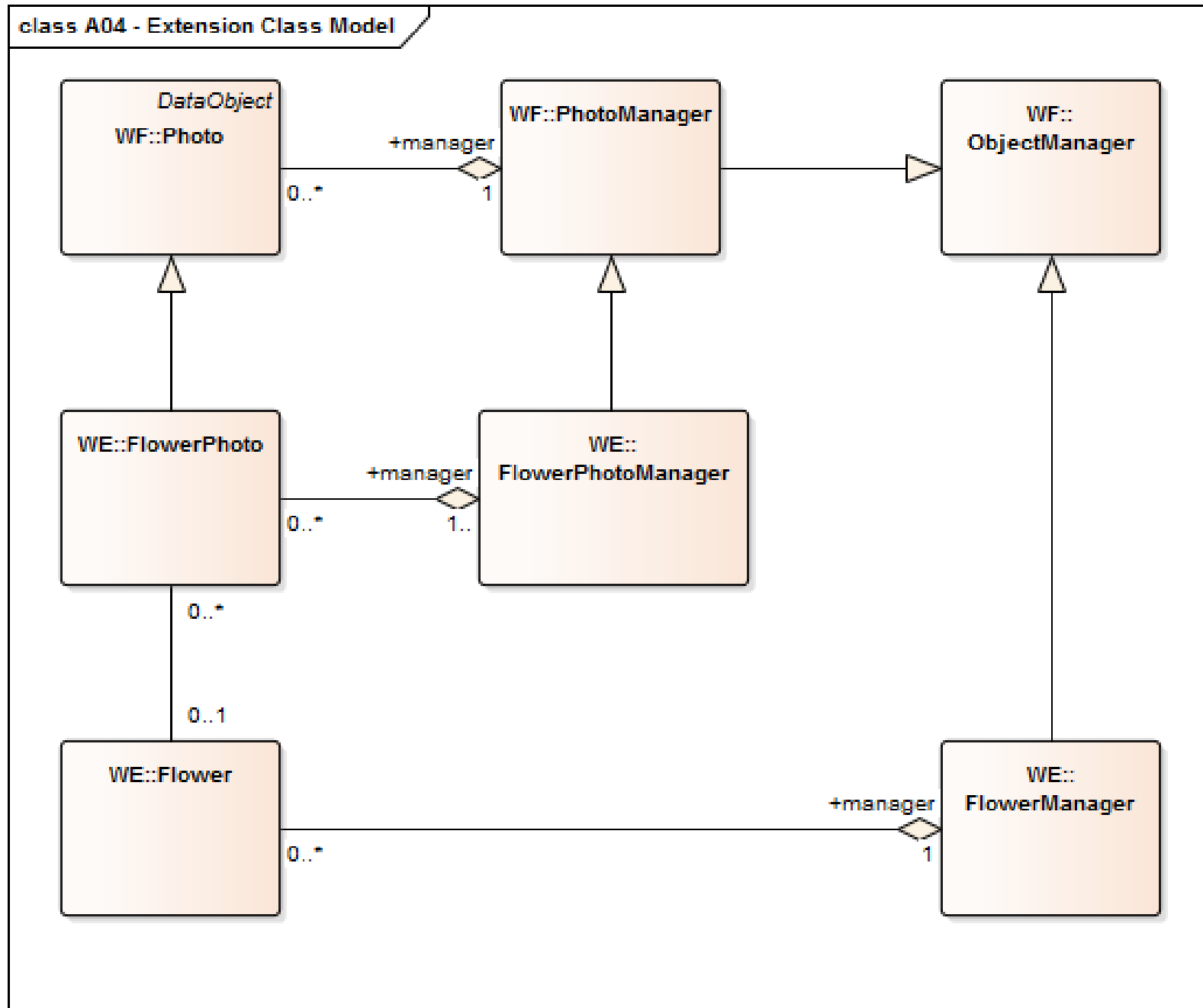
# Application Shutdown

- ShowAdminPageHandler (org.wahlzeit.handlers)
  - Administrators can request system shutdown
  - Handler will request ServiceMain to shut down
- Wahlzeit (org.wahlzeit.apps)
  - Listens to servlet container
  - Upon destroyed event, tells ServiceMain to shut down
- ServiceMain (org.wahlzeit.main)
  - Saves objects, closes database connections upon request

# Extending Wahlzeit

- Photo and PhotoFactory
  - At a minimum, these classes need to be extended for your domain
- Wahlzeit application startup
  - In addition, you need to create instances of your photo factory

# (Part of) Extending Wahlzeit Class Model



# Some Known Shortcomings

- A few @FIXME and @TODO annotations around
- User interface handling is simple but incomprehensible :-(
- A few too many singletons; can and should be simplified
- For more, see <https://github.com/dirkriehle/wahlzeit/issues>
- **Wahlzeit is not ready any longer for production use**

# Thanks! Questions?

**[dirk.riehle@fau.de](mailto:dirk.riehle@fau.de) – <http://osr.cs.fau.de>**

**[dirk@riehle.org](mailto:dirk@riehle.org) – <http://dirkriehle.com> – [@dirkriehle](#)**



# Credits and License

- Original version
  - © 2012-2018 [Dirk Riehle](#), some rights reserved
  - Licensed under [Creative Commons Attribution 4.0 International License](#)
- Contributions
  - ...

# Wahlzeit Design

**Prof. Dr. Dirk Riehle**  
**Friedrich-Alexander University Erlangen-Nürnberg**

## **ADAP A04**

Licensed under [CC BY 4.0 International](#)

It is Friedrich-Alexander University Erlangen-Nürnberg – FAU, in short.  
Corporate identity wants us to say “Friedrich-Alexander University”.

# Overall Architectural Style

- Servlet-based web application running on Google App Engine
- Persistence is managed by GAE using “Cloud Datastore”
- The architecture follows a typical model-view-separation style

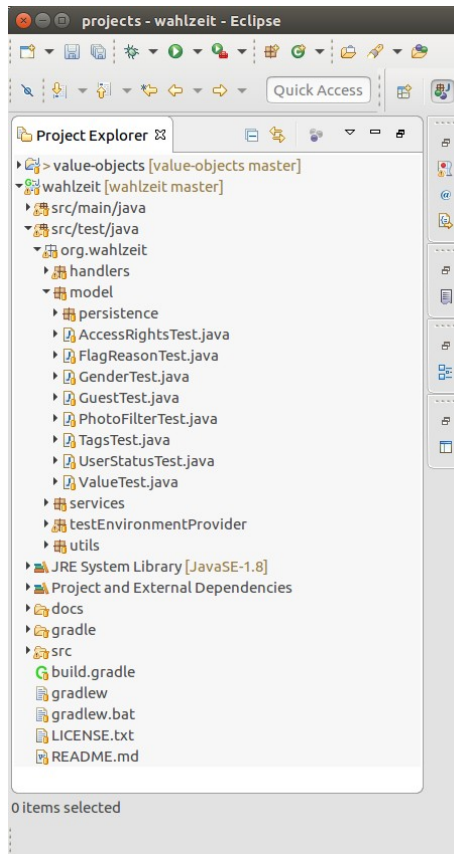
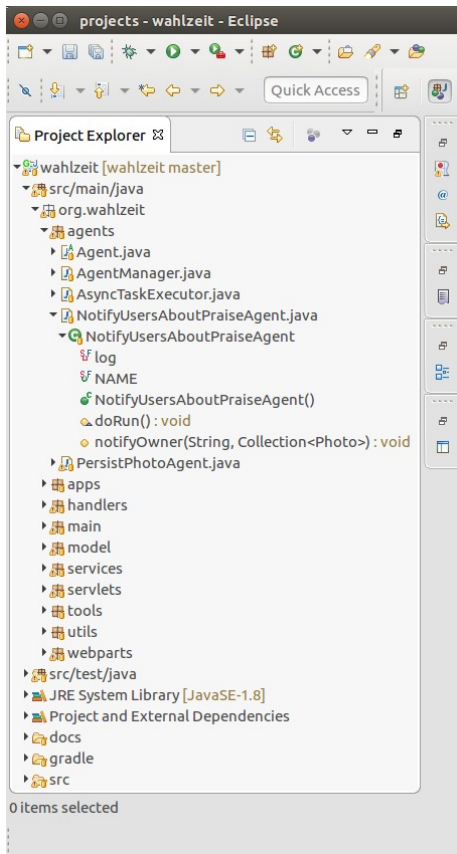
# “Physical” Architecture

- Client (browser)
- Application (application server)
- Database (database server)

# Application Architecture

- User interface (view)
- Domain model
- Database access

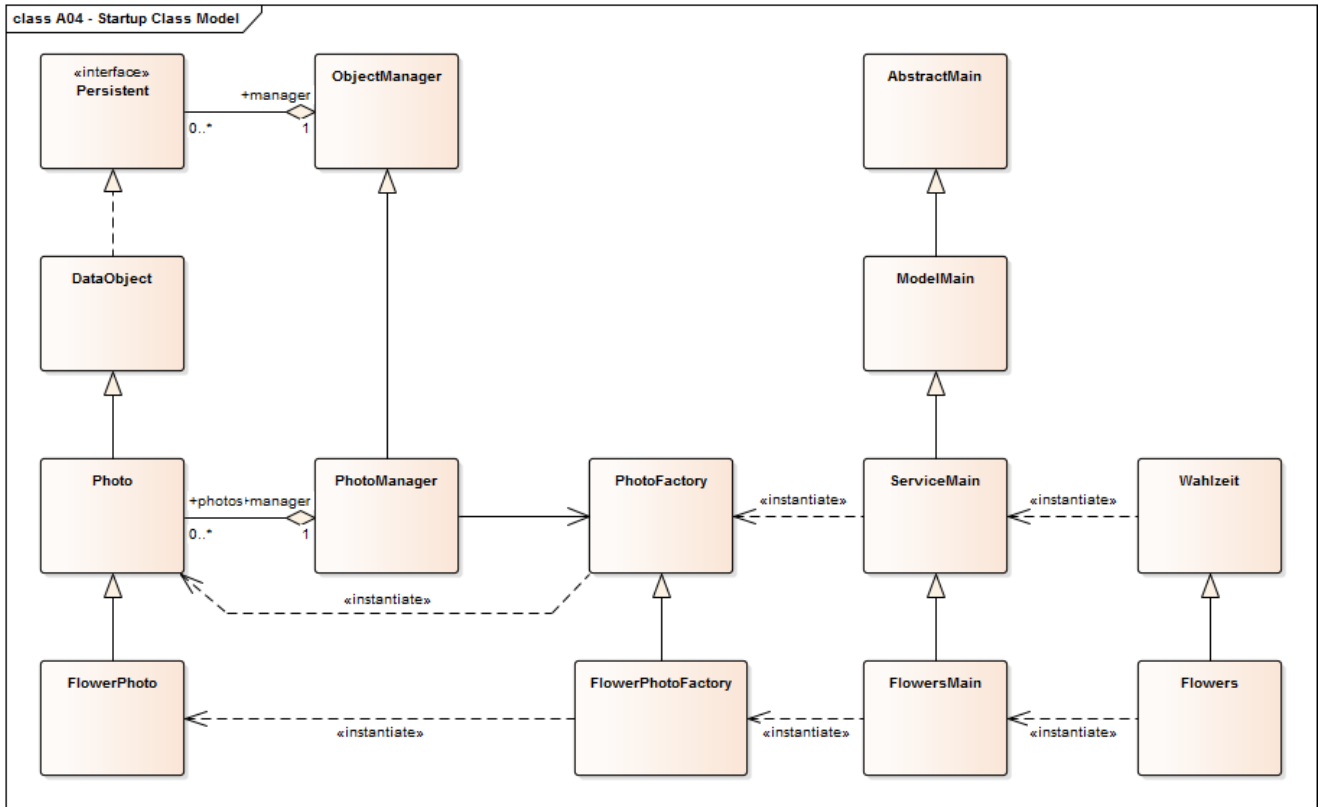
# Code Organization / Architecture



# Application Startup

- Wahlzeit (org.wahlzeit.apps)
  - A simple class hooked into the servlet container as listener
    - Listens to initialized and destroyed events
    - Delegates application startup and shutdown to Main class
- Main (org.wahlzeit.main)
  - A class hierarchy that handles system startup and shutdown
  - ServiceMain represents the main web service application
    - Starts-up all services; shuts them down upon request
  - ScriptMain provides a single faceless startup/shutdown cycle
    - Used to run scripts, for example, to migrate database schema

# Application Startup Class Model





# User Interaction

- Servlets (org.wahlzeit.servlets)
  - Default servlet handles static resource requests
  - Main Wahlzeit servlet handles all dynamic content requests
    - Has been configured with handlers per request type
    - Dispatches request to proper handler
- Handler (org.wahlzeit.handlers)
  - One handler for each page type of the application
    - Receives request information, performs handler-specific action
    - Knows how to call the domain model

# Web Page Generation

- **Handlers (org.wahlzeit.handlers)**
  - One handler for each page type of the application
    - Is configured with a web page (HTML) template
    - Requests templates and builds the web page
    - Utilizes a UserSession for a user's session state
- **WebParts and Templates (org.wahlzeit.webparts)**
  - A web-part is an object wrapping an HTML component
    - Is generated from a web-part template
  - All web-part templates are loaded from the file system
  - The WebPartTemplateService provides those templates

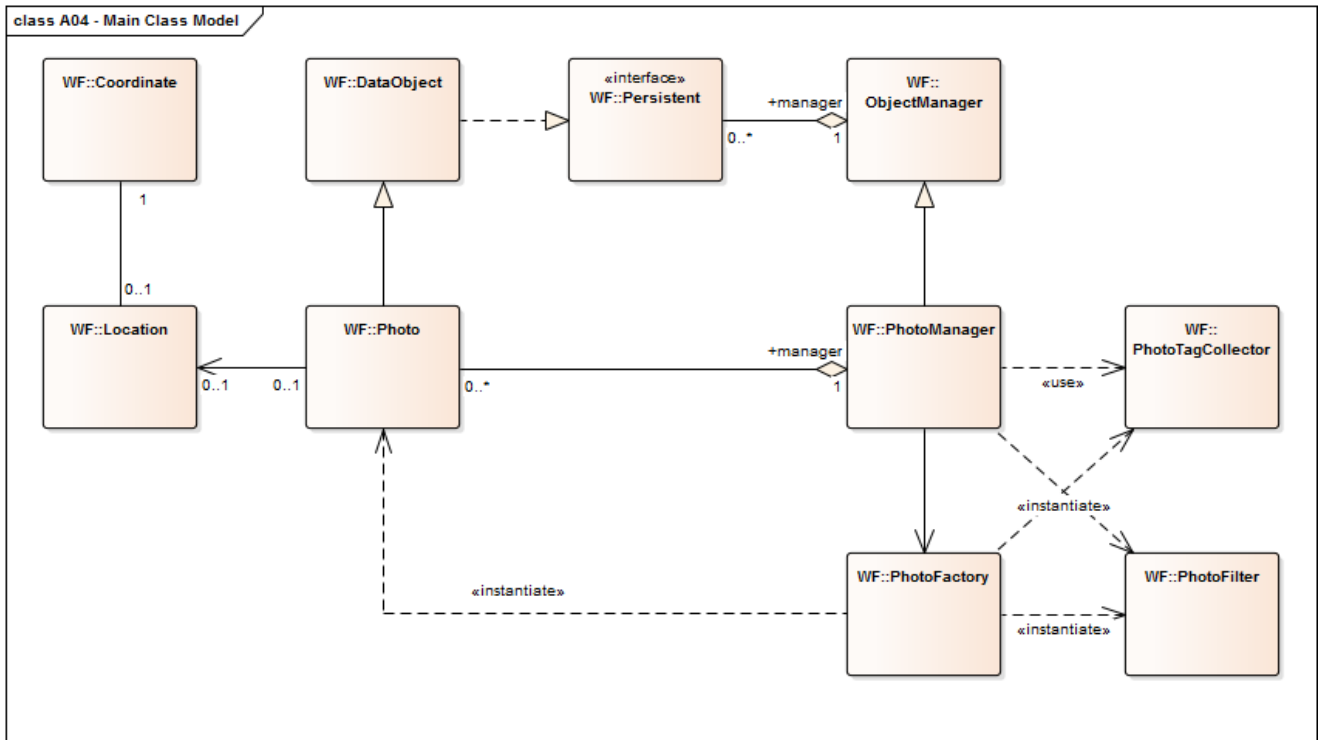
# Access Rights

- Client (org.wahlzeit.model)
  - Each visitor is represented by an object from the Client class hierarchy
    - Has appropriate AccessRights associated with it
- Handler (org.wahlzeit.handlers)
  - Each handler knows about the current user by way of the UserSession
    - Each access to a web page or a web-part is checked against user rights

# Domain Model

- Domain abstractions (org.wahlzeit.model)
  - Package contains all domain abstractions
    - Client hierarchy for visitors, logged-in users, moderators, administrators
    - Photo class hierarchy for photos
    - PhotoCase for cases filed against photos (copyright violation etc)
    - Various value object classes, e.g. language, status, gender
- Manager (org.wahlzeit.model)
  - An ObjectManager manages a particular domain abstraction (hierarchy)
    - UserManager manages the User class hierarchy objects, etc.
  - The object manager also hides the database access layer

# (Part of) Domain Class Model



# Object Persistence

- Persistent (org.wahlzeit.services)
  - Defines interface for reading and writing objects from storage
  - Provides dirty flag to indicate that object needs to be saved
- ObjectManager (org.wahlzeit.services)
  - Manages DataObject (implements Persistent) instances
    - One instance (of subclass) for each type of DataObject (subclass)
  - Encapsulates storage layer access

# System Configuration

- SysConfig (org.wahlzeit.services)
  - A key/value store with basic system configuration data
    - Provides key directories and URLs
- ModelConfig (org.wahlzeit.model)
  - A key/value store with model configuration information
    - Loaded from properties file
    - One for each language (currently only 'en' and 'de')
    - Provides language-specific text message formatting
- ConfigDir (org.wahlzeit.services)
  - Directory abstraction that provides custom extension mechanism

# Session Management

- Session (org.wahlzeit.services)
  - A session object maintains one database connection
  - For each active thread, one session object will be active
  - Session objects get allocated when thread becomes active
- SysSession (org.wahlzeit.services)
  - A session associated with faceless system threads
- UserSession (org.wahlzeit.services)
  - A session associated with each currently active user
  - Maintains user's session state, e.g. form input data



# Information Logging

- LogBuilder (org.wahlzeit.services)
  - A simple logging class that utilizes GAE logging

# Email Sending

- EmailService (org.wahlzeit.services.mailing)
  - Provides interface for sending email, various implementations:
    - SmtplibEmailService actually sends email
    - LoggingEmailService wraps email service and logs before sending
    - MockEmailService drops any request; needed for testing
- EmailServiceManager (org.wahlzeit.services.mailing)
  - Provides global email service instance
  - Can be configured for different execution scenarios (testing vs production)

# System Agents

- Agent (org.wahlzeit.agents)
  - An agent is a thread performing some faceless system functionality
    - Runs in the background, sleeps most of the time
- AgentManager (org.wahlzeit.agents)
  - Manages available agents, wakes them up

# Photo Praise Notification

- PraisePhotoFormHandler (org.wahlzeit.handlers)
  - If a photo owner requested praise notification, enqueues praise with agent
- NotifyAboutPraiseAgent (org.wahlzeit.agents)
  - Agent enqueues praise requests received from handler
    - Sends out praise information in regular intervals

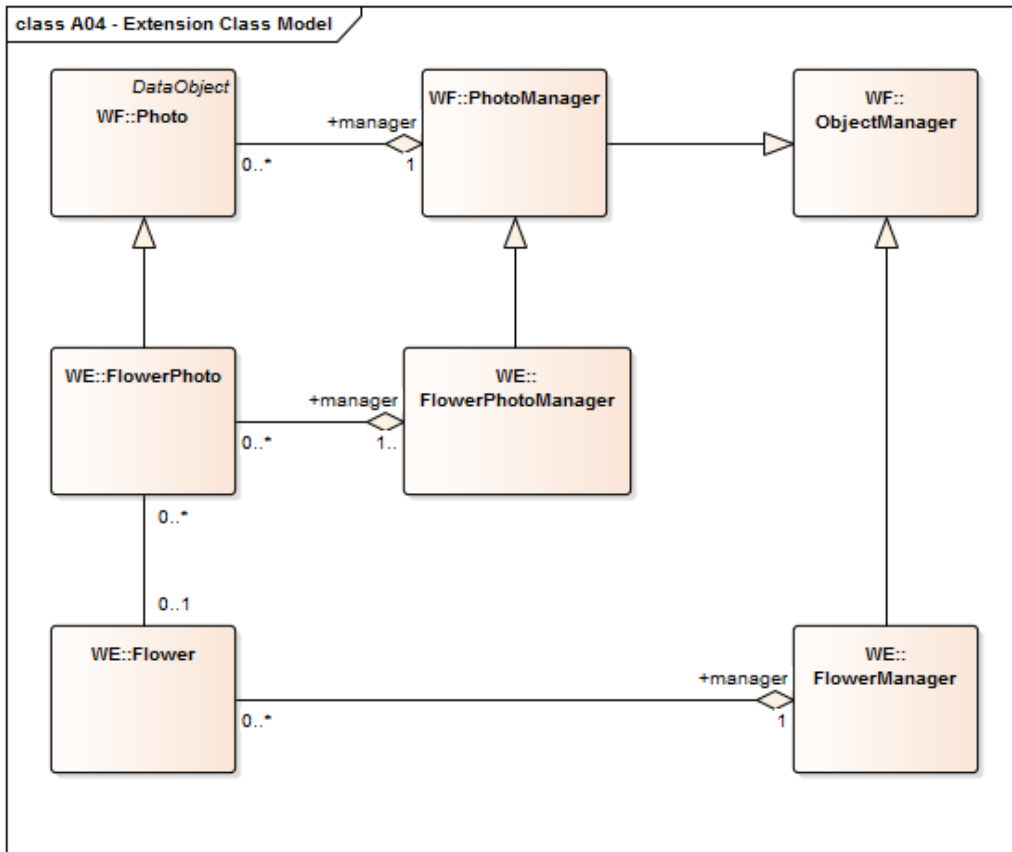
# Application Shutdown

- ShowAdminPageHandler (org.wahlzeit.handlers)
  - Administrators can request system shutdown
  - Handler will request ServiceMain to shut down
- Wahlzeit (org.wahlzeit.apps)
  - Listens to servlet container
  - Upon destroyed event, tells ServiceMain to shut down
- ServiceMain (org.wahlzeit.main)
  - Saves objects, closes database connections upon request

# Extending Wahlzeit

- Photo and PhotoFactory
  - At a minimum, these classes need to be extended for your domain
- Wahlzeit application startup
  - In addition, you need to create instances of your photo factory

# (Part of) Extending Wahlzeit Class Model



## Some Known Shortcomings

- A few @FIXME and @TODO annotations around
- User interface handling is simple but incomprehensible :-(
- A few too many singletons; can and should be simplified
- For more, see <https://github.com/dirkriehle/wahlzeit/issues>
- **Wahlzeit is not ready any longer for production use**



# Thanks! Questions?

**[dirk.riehle@fau.de](mailto:dirk.riehle@fau.de) – <http://osr.cs.fau.de>**

**[dirk@riehle.org](mailto:dirk@riehle.org) – <http://dirkriehle.com> – [@dirkriehle](#)**

DR

# Credits and License

- Original version
  - © 2012-2018 [Dirk Riehle](#), some rights reserved
  - Licensed under [Creative Commons Attribution 4.0 International License](#)
- Contributions
  - ...