Migrating to Neon

Judith Gull, Matthias Zimmermann

Agenda

What is Changing with Neon?

Migrating the «Contacts» Application

- ➔ Shared
- ➔ Server
- ➔ Client
- ➔ Demo

Migration Tips

Migration to Neon

What is changing?

What changes in Neon?

HTML-UI
Instead of RAP/Swing/SWT

Pure Java
Instead of OSGi Bundles

Other API changes

HTML-UI Changes

Goals: No conceptual changes in scout client model

- Possible to use existing models with minor modifications

Migration:

- Rewrite custom Swing/SWT/RAP components in JS/HTML
- New Attributes (CSS class)
- Fonts instead of Icons
- …

Vanilla Java

Challenge: Eclipse and OSGi code had to be removed

- No deprecations

Migration:

- Convert project structure to maven default
- Convert manifest dependencies to maven dependencies
- Convert product files to launch files
- Use bean manager instead of OSGi service registry, no service registration
- Eclipse Jobs -> scout jobs
- Test runner

Other API Changes

- Some renames/moves: AbstractExtensibleSmartField -> AbstractSmartField
- Improved Security: HTML is now escaped by default
- ClientNotification: Improved performance and API
- …

Migration Guide

- Not ready yet [1]

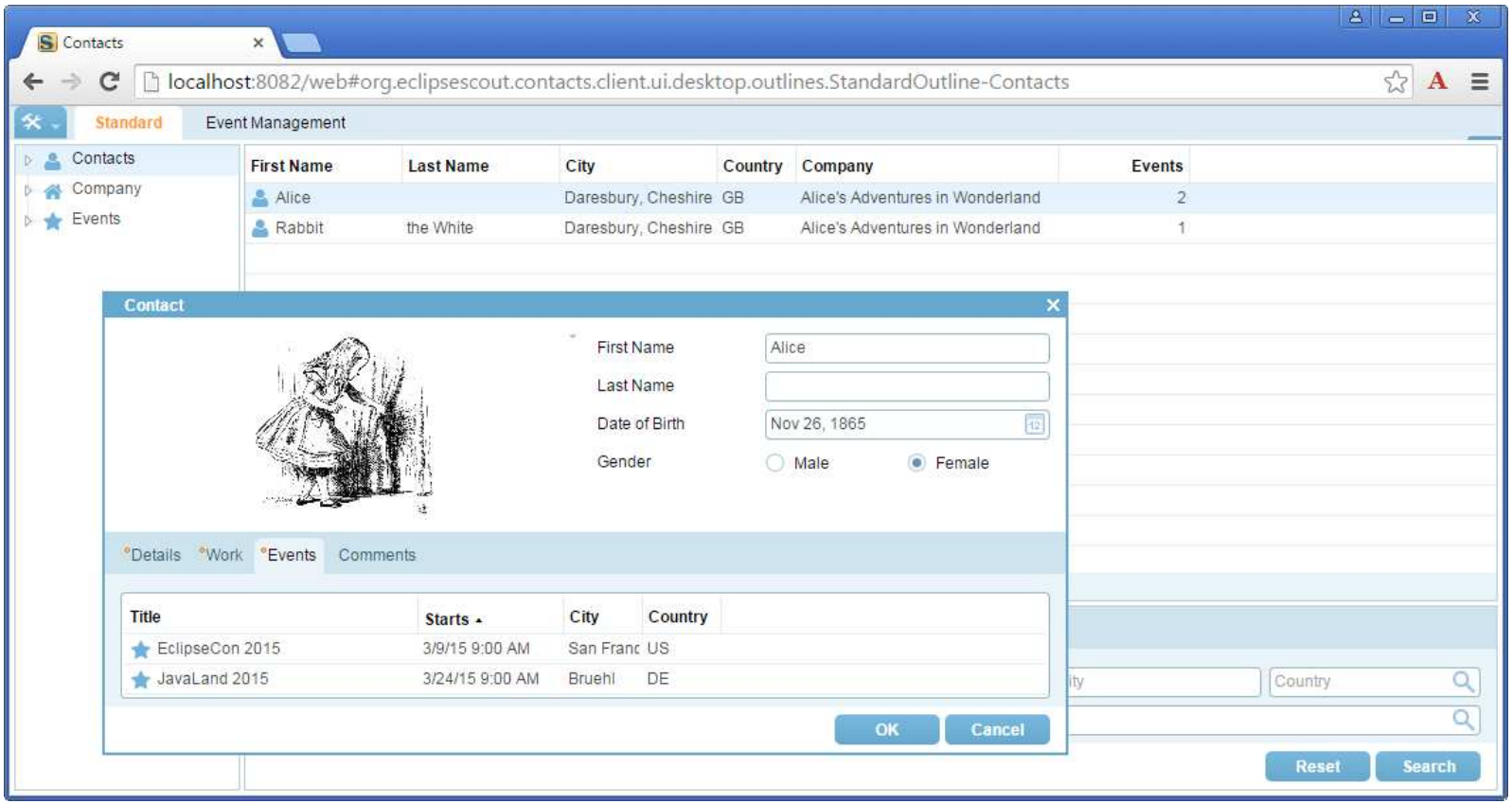
- Currently not much experience with migrations (BSI Standard Product)

- More projects expected to migrate in 2016

[1] https://github.com/BSI-Business-Systems-Integration-AG/org.eclipse.scout.docs/blob/releases/5.2.x/docs/adoc/migration/MigrationGuide.adoc

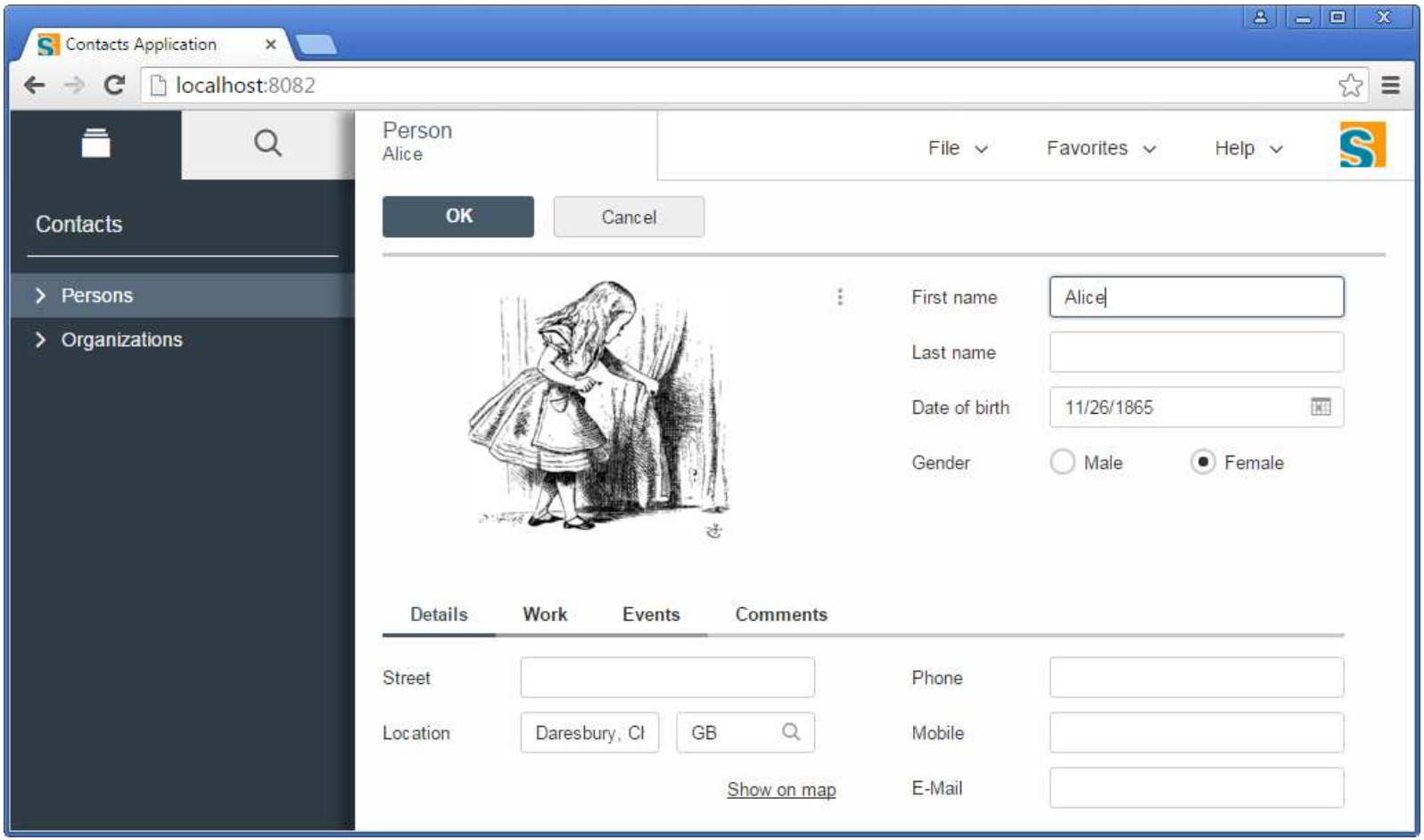
Migrating «Contacts»
A practical Example

«Contacts»
Mars



https://github.com/BSI-Business-Systems-Integration-AG/org.eclipse.scout.docs/tree/releases/5.0.x/code/contacts

«Contacts»
Neon



https://github.com/BSI-Business-Systems-Integration-AG/org.eclipse.scout.docs/tree/releases/5.2.x/code/contacts
https://github.com/BSI-Business-Systems-Integration-AG/org.eclipse.scout.docs/tree/releases/6.0.x/code/contacts

«Contacts»

Background

- ➔ Replaces Wiki «Minicrm» and Scout Book «My Contacts»
- ➔ Broad coverage of Scout features
- ➔ Showcase release migrations

Neon «Business» Updates

- ➔ Code organisation along business components
- ➔ Includes new features of HTML5 UI
  - Tabs replace modal forms
  - Tools
  - Theming

Migrating the «Contacts»

High level

1. Create a «Hello World» using the Scout SDK
2. Move existing business code to the new app
3. Remove errors and warnings

Lower level

➔ Migrate the core, then the dependent business modules

➔ Migrate a module

- First **shared**
- Then **server**
- Then **client**

Migrating «Contacts»
Shared

Migrating «Contacts»
org.eclipsescout.contacts.shared

Just copy & paste

- ➔ Permissions
- ➔ Codetypes & Codes
- ➔ Lookup calls
- ➔ FormData & PageData

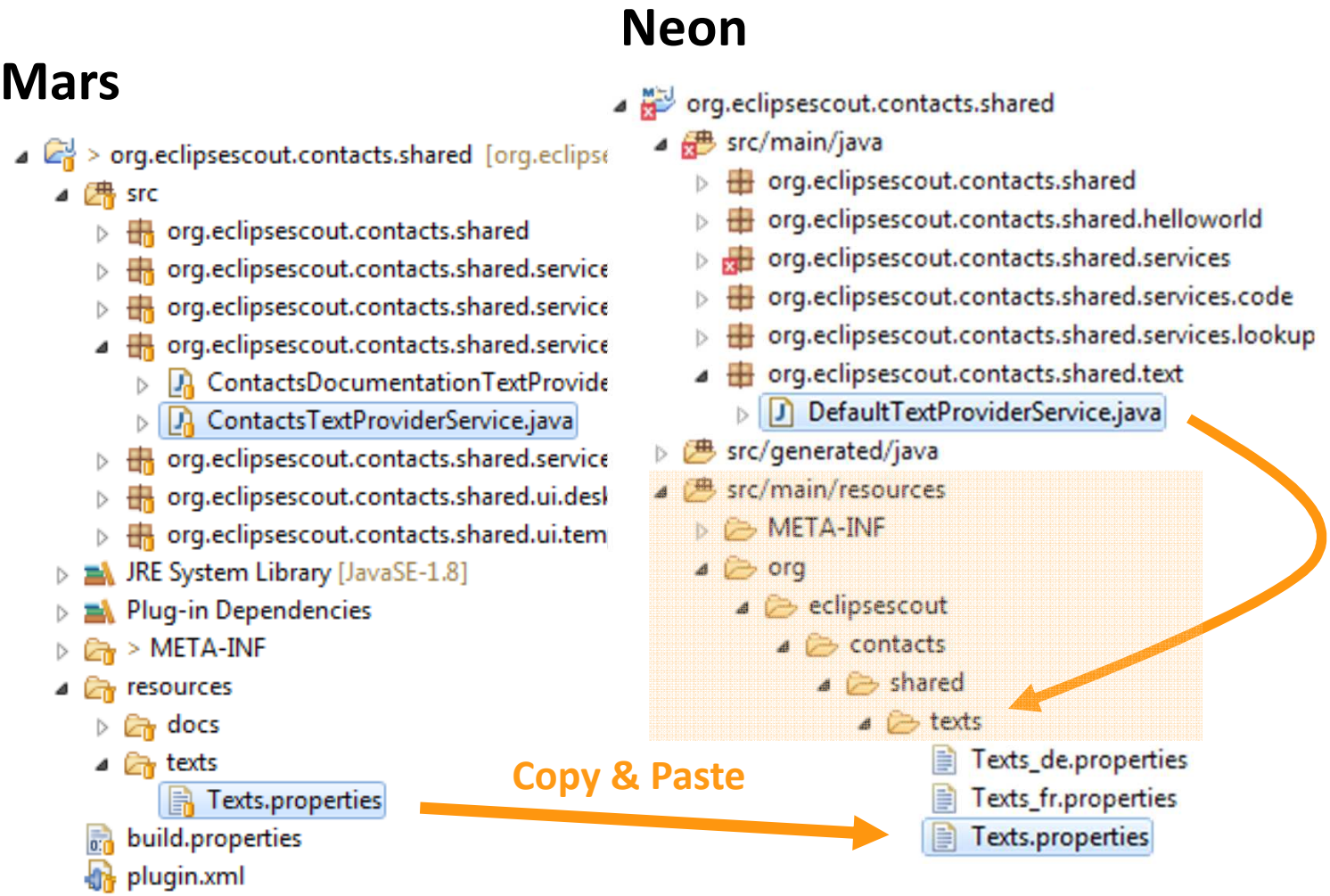
To modify for Neon

- ➔ Move Texts.properties to new place
- ➔ Lookup call interfaces
- ➔ Custom service interfaces

Skipping

- ➔ DefaultTextProviderService

Migrating «Contacts»
Texts.properties



Migrating «Contacts»
Lookup Call Interfaces

Mars

```
public interface IContactLookupService extends
ILookupService<String> { }
```

Neon

```
@TunnelToServer
public interface IContactLookupService extends
ILookupService<String> { }
```

Migrating «Contacts»
Custom Service Interfaces

Mars

```
@InputValidation(IValidationStrategy.PROCESS.class)
public interface IContactService extends IService {
   ...
}
```

Neon

```
@ApplicationScoped
@TunnelToServer
@InputValidation(IValidationStrategy.PROCESS.class)
public interface IContactService {
   ...
}
```

Migrating «Contacts»
Server

Migrating «Contacts»
org.eclipsescout.contacts.server

Just copy & paste

- ➔ Lookup services

To modify for Neon

- ➔ Add **ConfigProperties**, **SuperUserRunContextProvider**, **PlatformListener**
- ➔ Move content of config.ini to config.properties
- ➔ Move startup logic to PlatformListener
- ➔ DerbySqlService (pom.xml …)
- ➔ Custom services

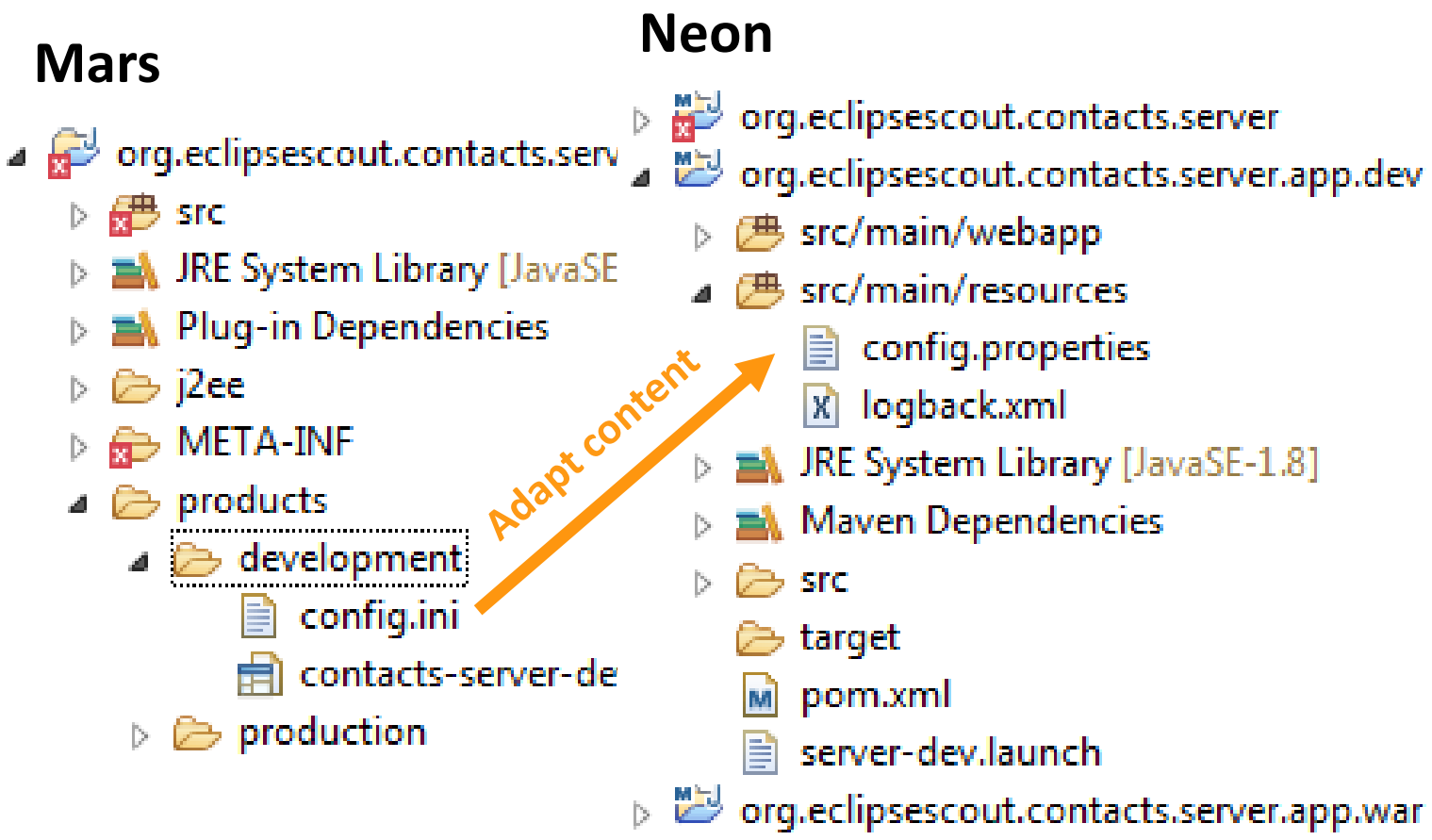
Skipping

- ➔ ServerSession
- ➔ AccessControlService

Migrating «Contacts»
Helper Classes (ConfigProperties …)

Neon

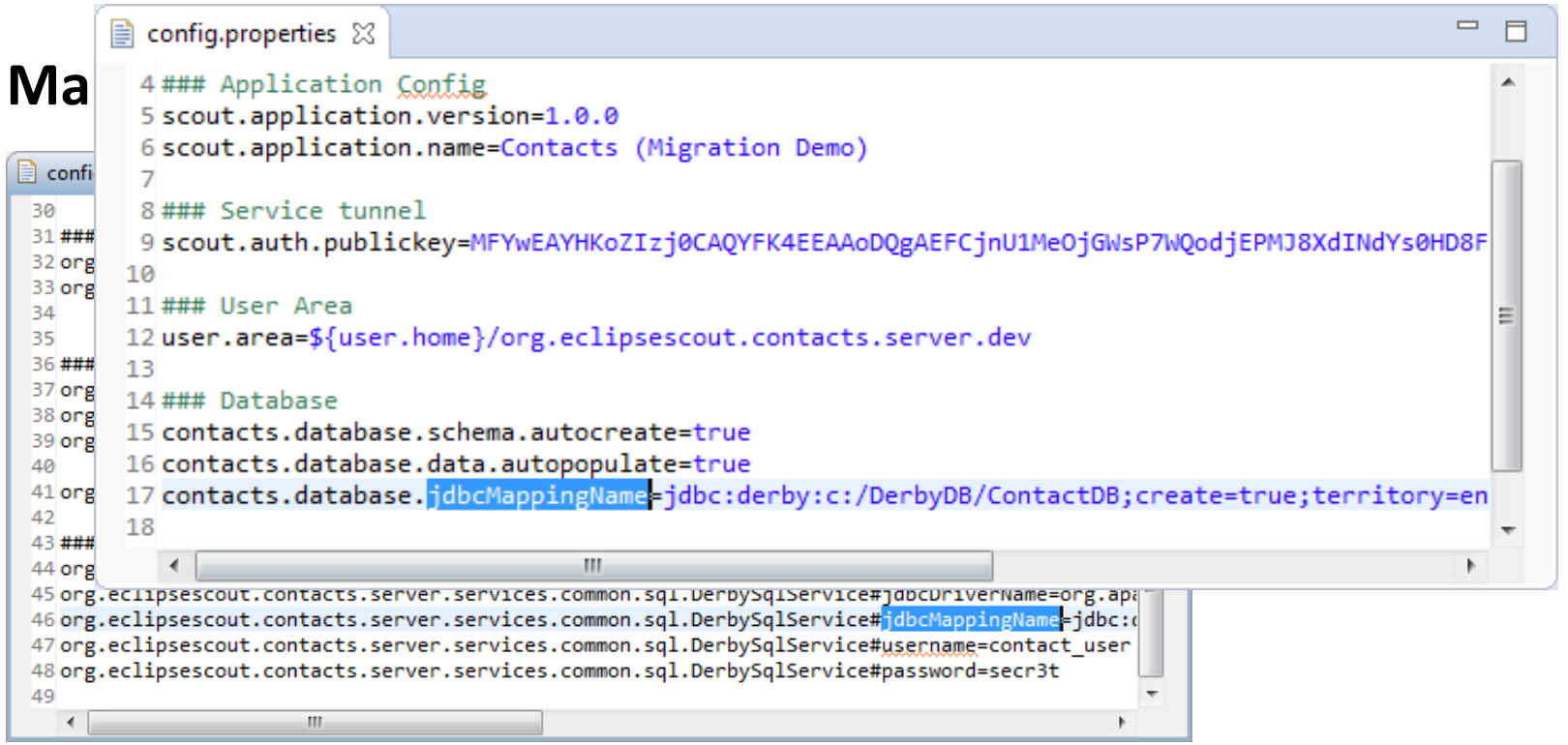
Migrating «Contacts»
Module per Product



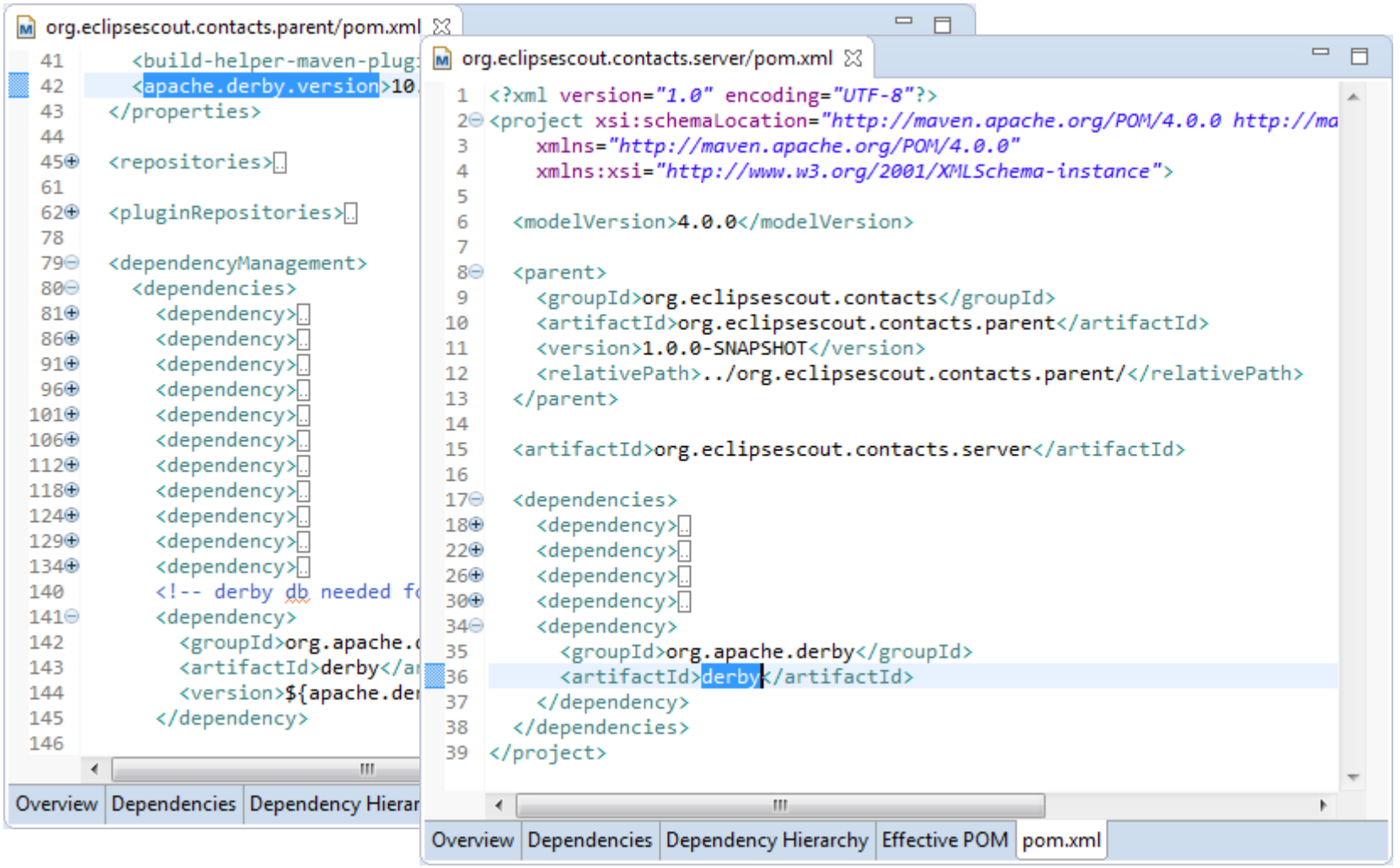
Migrating «Contacts»
config.ini – config.properties

Neon

Ma



Migrating «Contacts»
DerbySqlService – pom.xml



Migrating «Contacts»
DerbySqlService – ConfigProperties

Mars

```
public class DerbySqlService extends AbstractDerbySqlService {}
```

Neon

```
public class DerbySqlService extends AbstractDerbySqlService {

@Override
protected String getConfiguredJdbcMappingName() {
  return CONFIG.getPropertyValue(
    ConfigProperties.DatabaseJdbcMappingNameProperty.class);
  }
}
```

Migrating «Contacts»
Custom Services

Mars

```
public class ContactService extends AbstractService
  implements IContactService {
...
}
```

Neon

```
public class ContactService implements IContactService {
...
}
```

Migrating «Contacts»
ServerApplication (startup logic)

Mars

```
public class ServerApplication implements IApplication {

  @Override
  public Object start(IApplicationContext context) throws Exception {

      @Override
      public IStatus run(IProgressMonitor monitor) {
        // Call all startup services to collect all available extensions
        for (IServerStartupService service :
          SERVICES.getServices(IServerStartupService.class)) {
          service.init();
        }
      }

      return Status.OK_STATUS;
    }

    return EXIT_OK;
  }
}
```

Migrating «Contacts»
ServerApplication (startup logic)

Mars

```
public class ServerStartupService extends AbstractService
  implements IServerStartupService
{

  @Override
  public void init() throws ProcessingException {
    Set<String> tables = getExistingTables();
    createCompanyTable(tables, addInitialData());
    createContactTable(tables, addInitialData());

    LOG.info("DB install completed");
  }

}
```

Migrating «Contacts»
PlatformListener (startup logic)

Neon

```
public class PlatformListener implements IPlatformListener {
  @Override
  public void stateChanged(PlatformEvent event) {
    if (event.getState() == State.BeanManagerValid) {
      autoCreateDatabase();
    }
  }
  public void autoCreateDatabase() {
     ...superUserRunContext.run(new IRunnable() {

      @Override
      public void run() throws Exception {
        Set<String> tables = getExistingTables();
        createOrganizationTable(tables);
        createPersonTable(tables);
      }
    });
  }
}
```

Migrating «Contacts»
Client

Migrating «Contacts»
org.eclipsescout.contacts.client

Just copy & paste

- ➔ Lookup calls

To modify for Neon

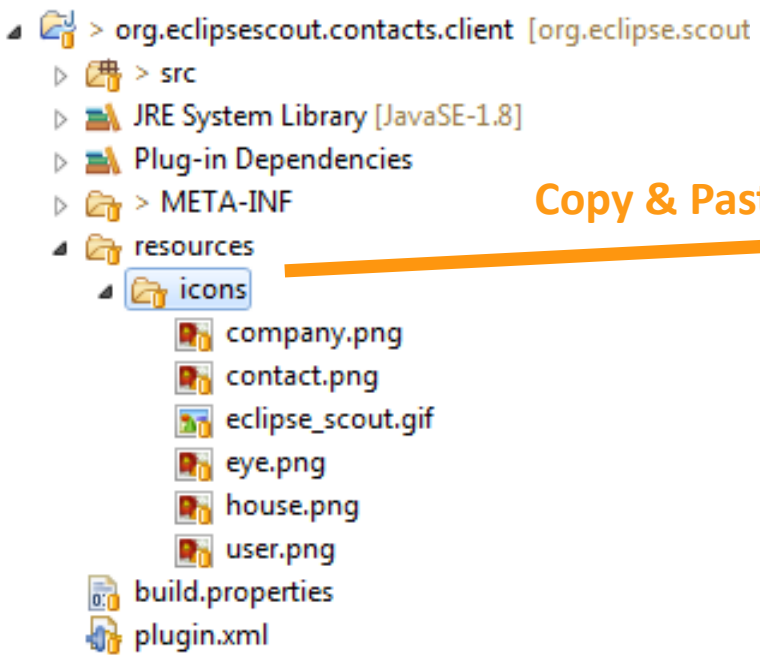
- ➔ Move Icon image files
- ➔ Forms & pages: «Pattern Matching»
  - AbstractExtensibleFoo → AbstractFoo
  - SERVICES.getService → BEANS.get

Skipping

- ➔ ClientSession

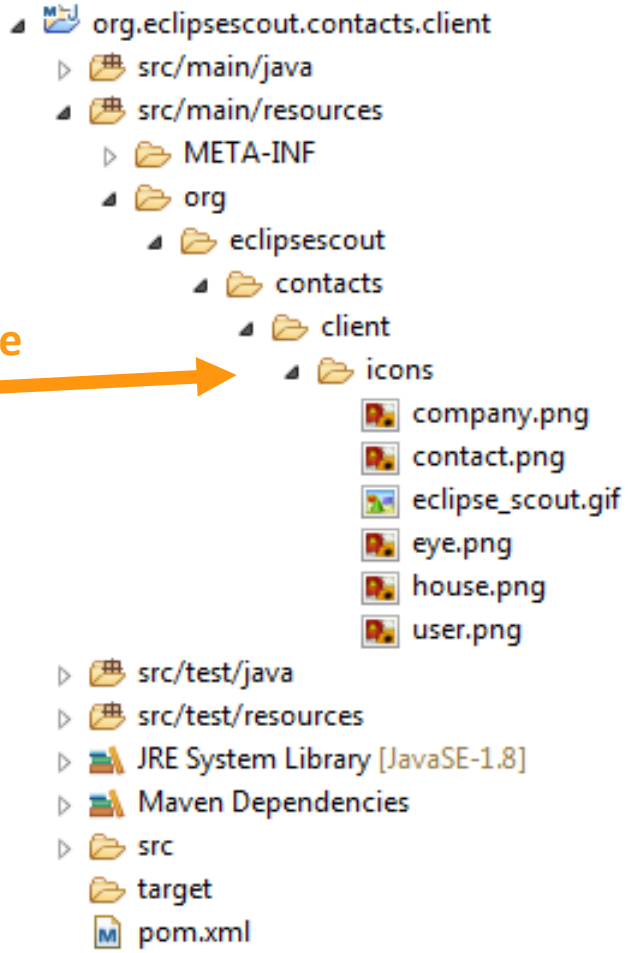
Migrating «Contacts»
Move Icon Images

Mars



Copy & Paste

Neon



Migrating «Contacts»
Client

Mars

```
@Order(1000.0)
public class EditURLMenu extends AbstractExtens

  @Override
  protected String getConfiguredText() {
    return TEXTS.get("EditURL");
  }
```

```
public class ModifyHandler extends AbstractFormH

  @Override
  protected void execLoad() throws ProcessingExc
    IContactService service = SERVICES.getServic
    ContactFormData formData = new ContactFormDa
    exportFormData(formData);
    formData = service.load(formData);
    importFormData(formData);
    setEnabledPermission(new UpdateContactPermis

  }
```

Neon

```
@Order(1000.0)
public class EditURLMenu extends AbstractMenu {

  @Override
  protected String getConfiguredText() {
    return TEXTS.get("EditURL");
  }
```

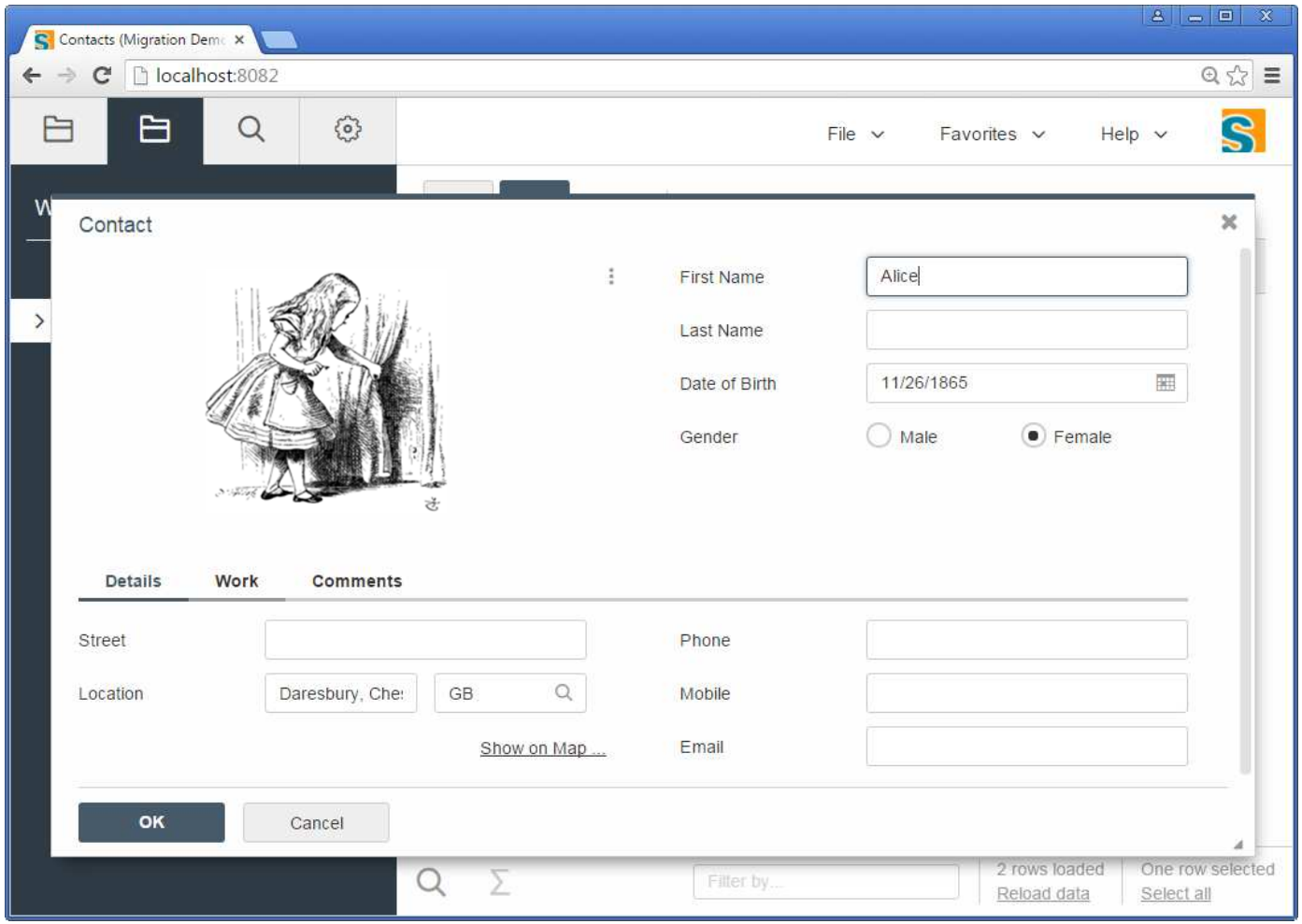
```
public class ModifyHandler extends AbstractFormHandler {

  @Override
  protected void execLoad() throws ProcessingException {
    IContactService service = BEANS.get(IContactService.class);
    ContactFormData formData = new ContactFormData();
    exportFormData(formData);
    formData = service.load(formData);
    importFormData(formData);
    setEnabledPermission(new UpdateContactPermission());

  }
```

Migrating «Contacts»
Demo

«Contacts»
Migration Demo



Migration to Neon
Tips

Test and Check Early

- Play with new sources early
- Are all your essential features implemented?

Plan Ahead

- Team may not be able to work as usual during migration
- Merges may be more difficult (project structure)

Plan Knowhow Transfer

- Be aware of new concepts in Scout Neon!
  - Check out the docs (book, forum) [1]
  - BSI Offers 2 day Upgrade Course (Mars -> Neon)

[1]https://tools.bsiag.com/scoutbook/5.2/latest/book_scout_architecture/scout_architecture/scout_architecture.html

Need Help?

- BSI offers migration support

Summary

Migrating to Neon
Summary

«Contacts»

- ➔ Technical migration, straight forward
- ➔ No change for the majority of components
- ➔ Includes refactoring, usage of new features

General:

- ➔ Positive feedback from migration of BSI CRM to Neon
- ➔ Migration tooling depending on additional experience
- ➔ We are here to help ☺

Thanks

@EclipseScout

{judith.gull | matthias.zimmermann}@bsi-software.com