

exe

Application Infrastructure
framework for web and
desktop **migrateble**
XQuery applications



Goals

Install a number of eXist applications on a same box, with a click of a button.

- Developers to create new applications quickly and efficiently.

- Blank slate syndrome
- Code and Solutions Reuse
- Create Code Repository

- End users to install eXist applications without need of any specific technical knowledge.

- Desktop or Web server
- Much bigger audience

What is a migratable application?

- MA installs from a .zip package or from a directory.
- During the installation of MA the framework:
 - Imports code, data and resources (js, css, images).
 - Creates collections, user and groups.
 - Sets up the security for the application pages.
 - Creates triggers and indexes.
- MA can pack all its code, current data & security.
- Packed MA can migrate back and forth between boxes with a single command.

Framework concept

1. Infrastructure:

- Installation, configuration and packaging
- Static resources management: JS,CSS, Images.

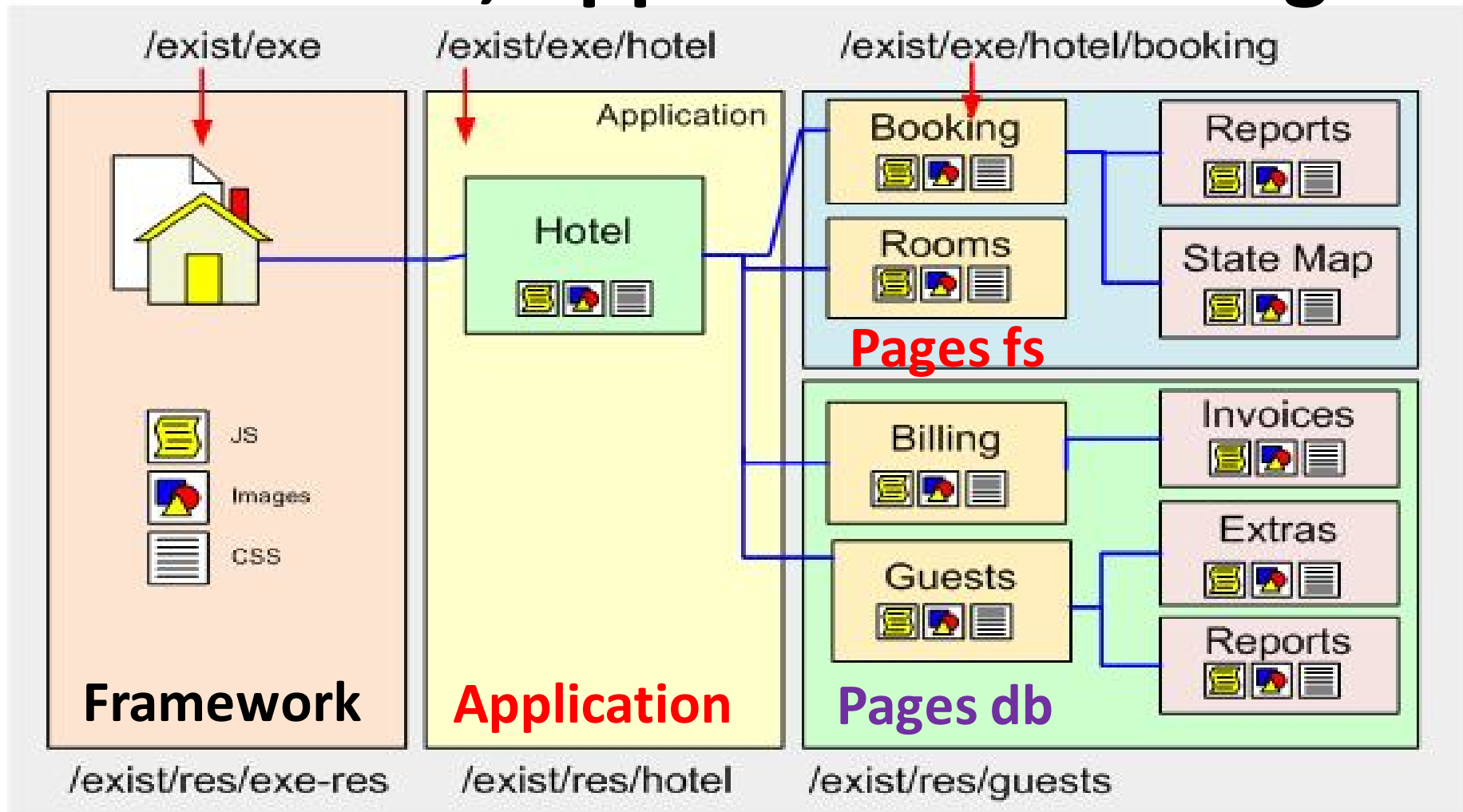
2. Common tasks and content:

- Navigation and URL-rewriting for to the relevant page
- Manage access to the pages of the applications
- User management: login, registration, forgotten pswd.

3. Content - applications and pages:

- An application is a hierarchical set of pages.
- A page is a collection of code and static resources to produce content. One page can be used many times.

Framework, Applications and Pages



- No limit in levels of page hierarchy within an application
- No limit to number of applications

eXe in Action - Framework controller

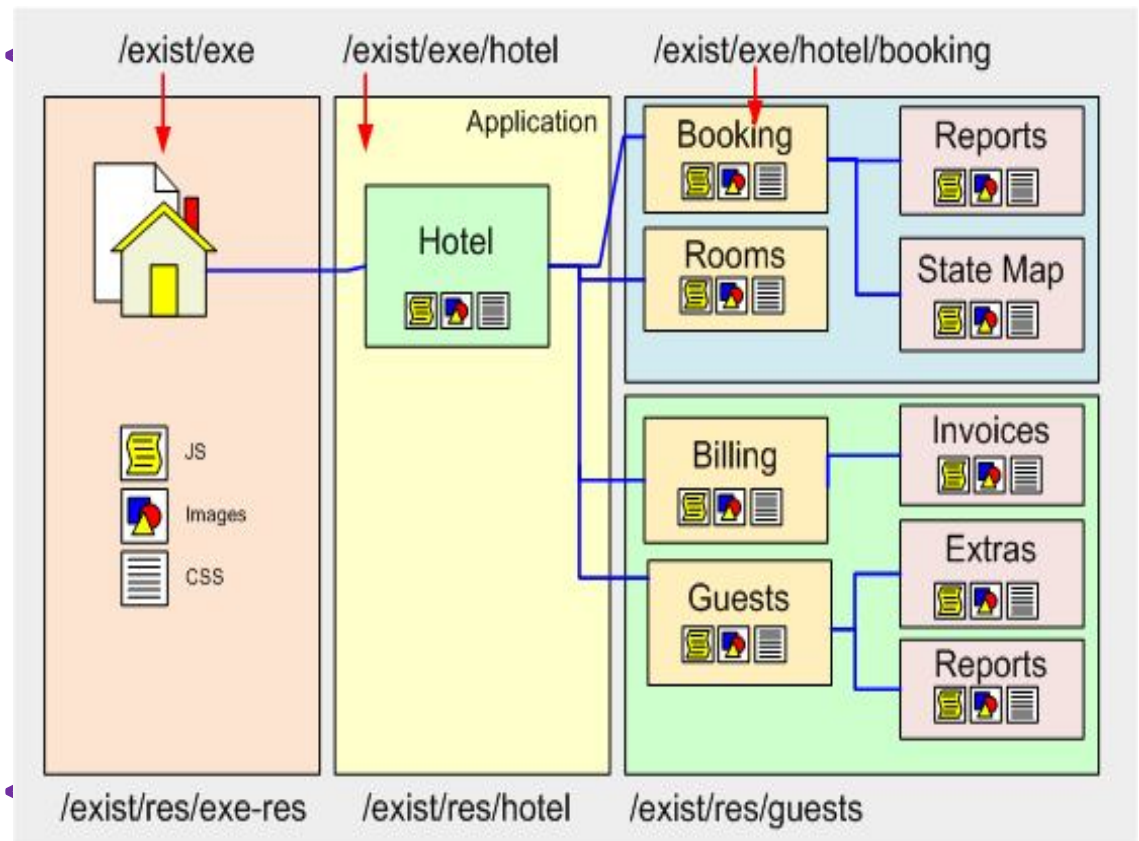
1. Allocates the application and the page.
2. Check access to every page in the path
3. Saves common data as request attribute:
 - Configuration: Framework, Application, Page
 - Navigation: Menu level 1, 2, 3, Crumb Trail
 - Page Head: Title, CSS, JS libraries, Meta tags
 - Bottom page resources: other JS
 - Footer
4. Forwards to a specific XQuery file in the page collection or directory.

eXe in Action – Page content

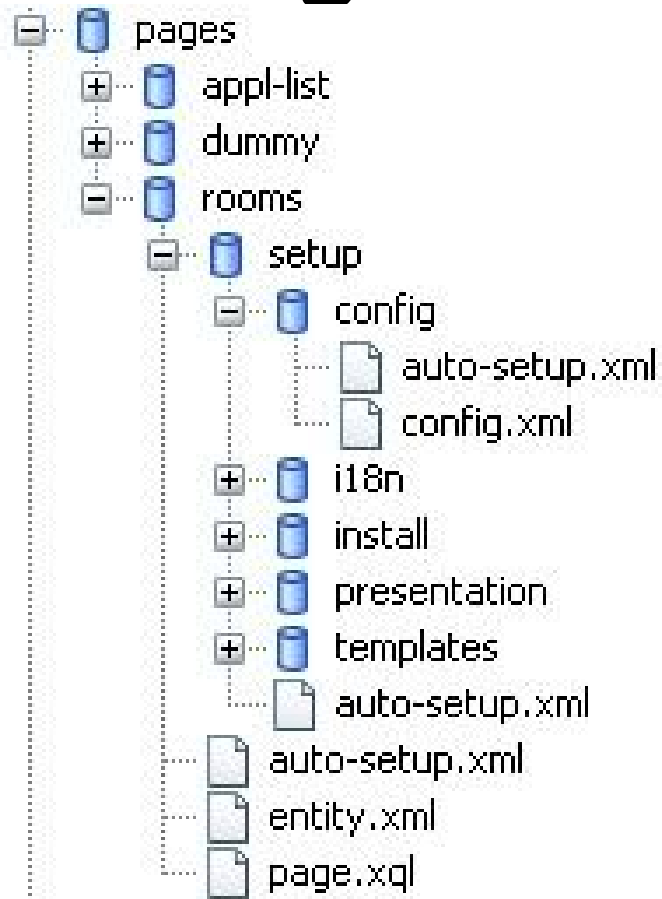
1. The page is a simple XQuery that does one thing.
2. Fills the blanks in a page template.
3. Uses saved navigation or creates its own.
4. Configuration data contains URIs to all relevant data collections.

eXe in Action – Resources

1. Every page has resources list files: JS, CSS
2. Resource files are listed automatically.
3. A resource has:
 - depends-on
 - library
 - disabled
 - external
4. Resource order:
 - framework
 - application
 - page.

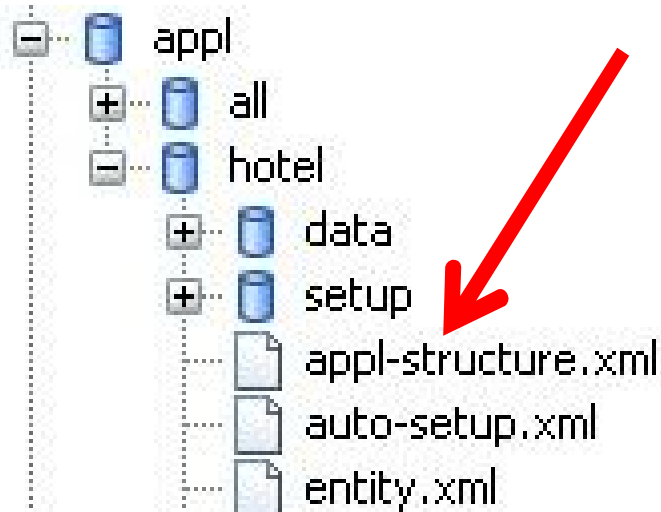


eXe Page Data Structure



```
<entity>
  <type>page</type>
  <id>rooms</id>
  <title>Rooms</title>
  <icon/>
</entity>
```

eXe Application Data Structure

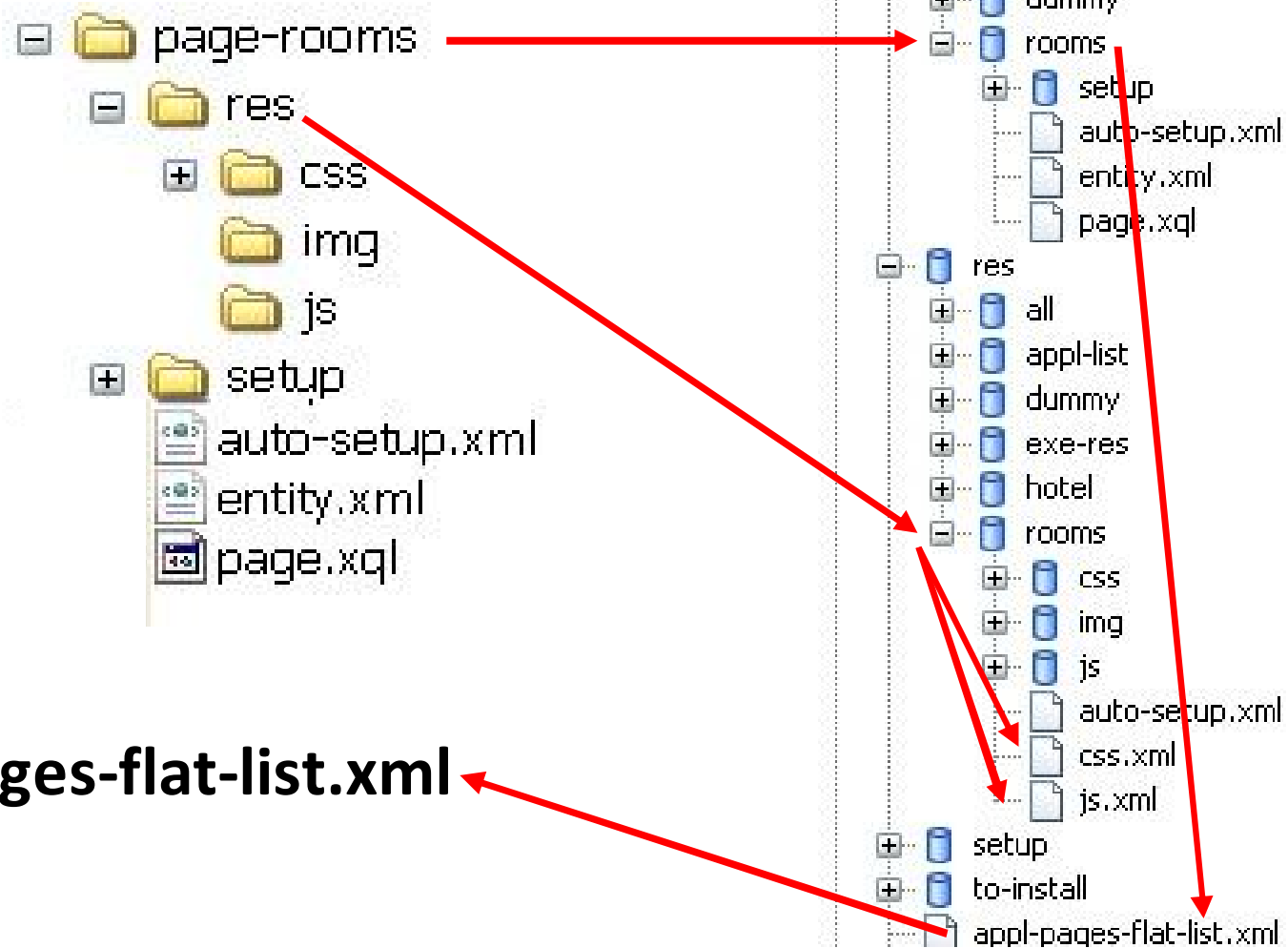


```
<entity>
  <id>hotel</id>
  <type>application</type>
  <title>Hotel Check In</title>
  <code-location>fs</code-location>
</entity>
```

```
<page>
  <id>clients</id>
  <parent>application</parent>
  <title>Rooms</title>
  <page-id>rooms</page-id>
  <icon/>
  <file>client-in-rooms.xql</pos>
</page>
```

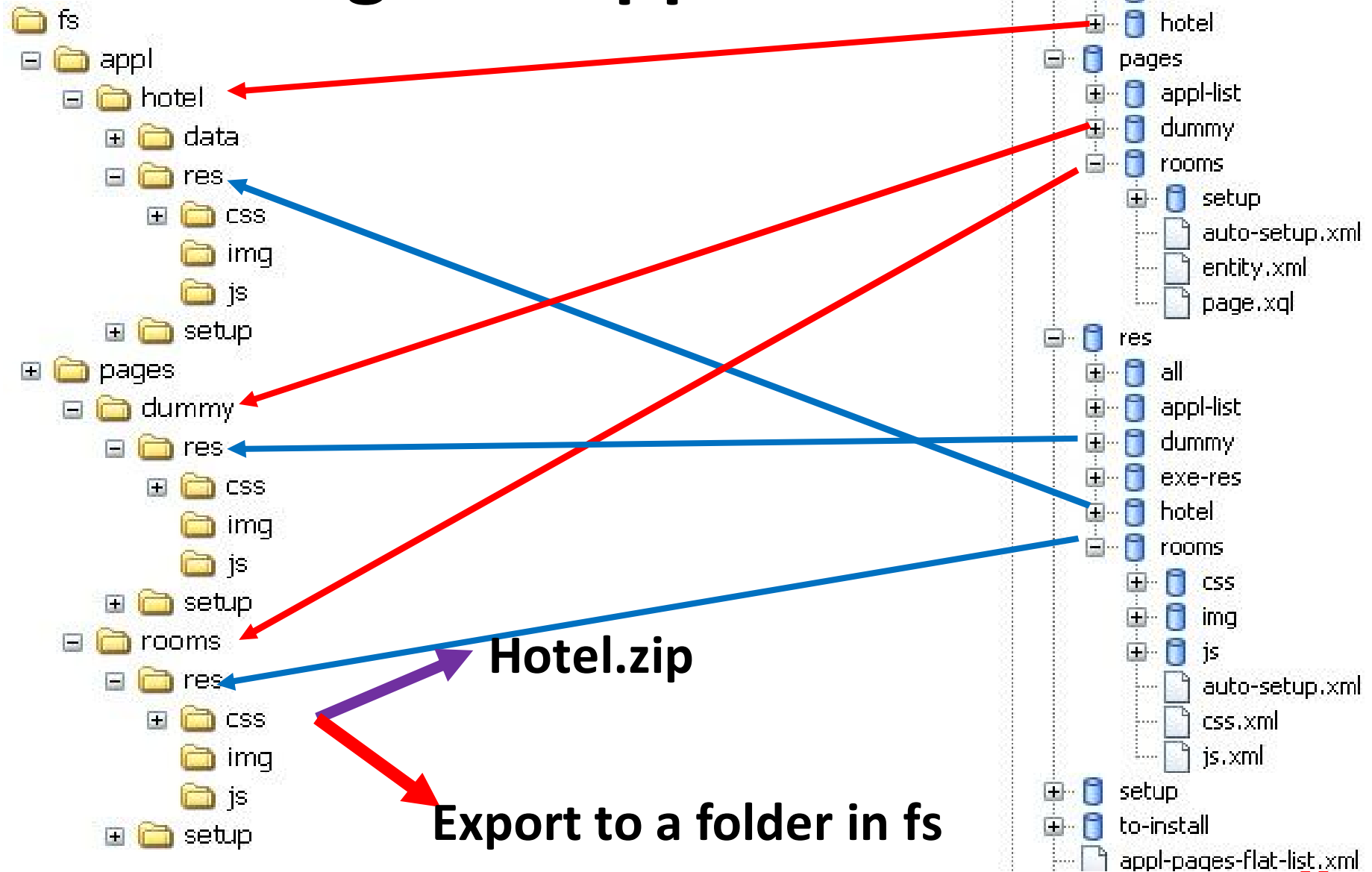
```
<page>
  <id>rooms</id>
  <parent>application</parent>
  <title>Rooms</title>
  <page-id>rooms</page-id>
  <icon/>
  <file>page.xql</pos>
</page>
```

eXe Page Installation



appl-pages-flat-list.xml

eXe Package an Application



eXe General Notes

- All this with web interface:
 - Install, Pack, fs \leftrightarrow db migration
 - Security
 - Resources management
 - Skins
- A page/application can migrate between db and fs
- IDs has to be unique, name spaces + versions
- Combine multiple JS and CSS in a single request.
- Encourage reusability of the pages

eXe Application Framework

Questions

