Concept - Core and jaas split

Problems

- remoting to Magnolia is nearly impossible
- move to JCR security in 4.5 left core intertwined with jaas module
- testing with security enabled is currently impossible
- replacing jackrabbit with other JCR implementation is not possible

Goals

- Provide Interface only module for RMI/JNDI access
- Separate JCR implementation from core.
- Allow testing with security enabled (not possible since 4.5)

Current Status

- there's 115 interfaces in core currently
    - not all would need to go, but most should
- 17 classes accessing JR API directly in core
    - excluding `jcr.commons`, but
    - including `org.apache.jackrabbit.JcrConstants`
- 2 classes accessing JR API directly in jaas

Open Questions / Discussion

API/Interfaces module

- **keep the package or change the package and have original interface still in core extending the new one?**
    - having same package in multiple jars is confusing
    - having multiple extends complicates hierarchy and would eventually end up in some conflict
      decision: change packages, do not provide extends, instead provide ppl with guide on what interface changed to what
        - **What is new package name? *info.magnolia.api***
- **do we restrict it to interfaces only?**
    - commonly used classes need to be accessible to `<jcr-impl>-provider` module which should not depend on core
    - core test jar have to depend on "some" (JR?) `<jcr-impl>-provider` to allow for security testing in core
    - if this module contains implementations as well as interfaces it will be useless for RMI/JNDI purposes

Content of the new Module (if we move classes along the interfaces):

- *interface*/classes
    - *Context*
        - *SystemContext*
            - *ThreadDependentSystemContext*
        - *UserContext*
        - *WebContext*
    - MgnlException
    - *BeanMerger*
        - **just the interface or also impl?**
    - Path
        - getUniqueLabel methods talking HierarchyManager and Content should stay in core or dropped completely
        - in use by jackrabbit-provider
    - MgnlNodeType
    - *ACL*
    - Permission
        - in use by jackrabbit-provider
    - UrlPattern
        - in use by jackrabbit-provider
    - *AccessManager*

- AccessManagerImpl
    - in use by jackrabbit-provider
- ContentRepository
    - in use by jackrabbit-provider
- RepositoryConstants
- *VirtualURIMapping*
- SystemProperty
- i.m.repository
    - *RepositoryManager*
    - RepositoryConstants
    - RepositoryMapping
    - WorkspaceMapping
    - DefaultRepositoryManager (should stay in core)
    - *Provider*
    - RepositoryDefinition
    - RepositoryMappingDefinition
    - RepositoryMappingDefinitionReader
    - WorkspaceMappingDefinition

jackrabbit-provider-module

- Currently JR deps are in core, but security impl is in jaas. That makes it impossible to run RepoTestCase w/ security enabled
- Direct JR dependency also makes it impossible to completely replace JR w/ other JCR impl
- JR-Provider module would consist of
    - `ProviderImpl` class
- few packages & few classes already deprecated in anticipation of the split
    - `MagnoliaAccessManager`
    - `MagnoliaAccessProvider`
    - `MagnoliaACLEditor`
- dependencies (that currently exist in core and are in use by jackrabbit related code)
    - security related
        - `AccessManager`
        - `AccessManagerImpl`
        - `Permission`
        - `UrlPattern`
        - `ACL`
    - basic utils
        - `Path`
        - `SystemProperty`
        - `ContentRepository`
            - **should not be used !!!!**
        - `i.m.repository.*`
        - `LifetimeJCRSession`
- All classes above except `ProviderImpl` are deprecated and could be moved
- All classes here incl. `ProviderImpl` are used by us only and not extended by anyone
- **Move to separate module**

magnolia-jaas

- Dependencies on (current) core:
    - *PrincipalList*
    - *ACL*
    - ACLImpl
    - ~~*Entity*~~
    - *GroupList*
    - *PrincipalCollection*
    - PrincipalCollectionImpl
    - *RoleList*
    - *Realm*
    - RealmCallback
    - BooleanUtil (umm, really???)
    - *User*
    - SecurityUtil
    - MgnlUser

    - MgnlUserManager
    - SecuritySupport
    - *UserManager*
    - *jackrabbit.ValueFactoryImpl*
- JackrabbitAuthenticationModule
    - should move to `jackrabbit-provider-module`
    - dependencies
        - ~~ContentRepository~~
        - PrincipalUtil
        - *User*
        - *UserManager*
        - *Context*
        - MgnlContext
        - *MagnoliaConfigurationProperties*
        - Components
- MagnoliaAuthenticationModule
    - is this used anywhere?
- MagnoliaJRAdminPrincipal
    - should move to `jackrabbit-provider-module`

others

i.m.beanCoder

- MgnlNode, MgnlProperty, MgnlValue
- used by workflow
- **should go w/ replacement or be moved in m_workflow**Channel
- should get accompanied by Site, Theme, Domain from STK

VirtualURIMapping

- 5 different impls under info.magnolia.cms.beans.config
- **should get their own package**
- maybe whole VURIM should get own module to be allowed develop further (support for virtual hierarchies discussed in forum more then once)jcr.registry
- *SessionProvider*, DefaultSessionProvider, SessionProviderRegistry
- doesn't seem to be used at all
- we seem to use i.m.repository.* instead
- **get rid of it?**

split core-compatibility from core

- *HierarchyManager
- *NodeData
- Query*
- etc.

just a jar, not module

- **what is the benefit?**
- **is it worth the effort?**

Proposal

- Split API/kernel module from core
    - this module should contain all the interfaces necessary for starting up the repo and common classes necessary for startup (e.g. Listener registered in web app)
- `jndi` or `<jcr-impl>-provider` modules should contain concrete implementations necessary for startup of the repo with given JCR implementation or by connecting to remote repository
- `magnolia-jaas` module should contain all security related interface implementations and utilitites
- renaming
    - jaas module should be renamed to `magnolia-security` (please note this is just a maven module, not magnolia module)
    - ideally `magnolia-core` would be renamed to `magnolia-cms-core` to signify better its purpose - having core running gives you naked cms engine, while other "modules" are just infrastructure for the cms engine
- jackrabbit related code should be moved from core and jaas into `jackrabbit-provider-module` (should not be magnolia module)
- (maven) dependencies resolution
    - API/kernel doesn't depend on anything
    - jcr-impl provider depends on API/kernel
    - magnolia-security depends on API/kernel

    - core depends on API/Kernel
    - nether of latter 3 modules depend on each other. Resolution of concrete implementations happens at runtime
    - core-test jar would depend on jackrabbit-provider-module and magnolia-security (means all our tests run against jackrabbit, with security fully configured)

Comments from 1st review

- Philipp: dependency (or lack thereof) between core and jackrabbit-provider seems to be backwards and doesn't feel right

Comments from 2nd review

- API stays minimalistic and interfaces only
- in the future plan split of core to core and kernel (startup, ioc, also minimalistic)
- where will be the tests and RepoTestCase is still the question, prefer to stay in core/kernel
- Next steps:
    - start preparatory work
    - next review after exploratory branch builds up and starts up

Followup tasks

- magnolia-jndi review & rework
- modeshape-provider-module reimplementation for 4.5