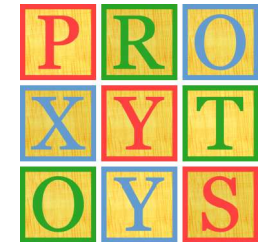


Developing loosely coupled systems with
Dependency Injection,
PicoContainer, NanoContainer and AOP

Aslak Hellesøy

Senior Developer - ThoughtWorks

My geekwork



Agenda

- Dependency Injection
- PicoContainer
- NanoContainer
- Questions/Discussion

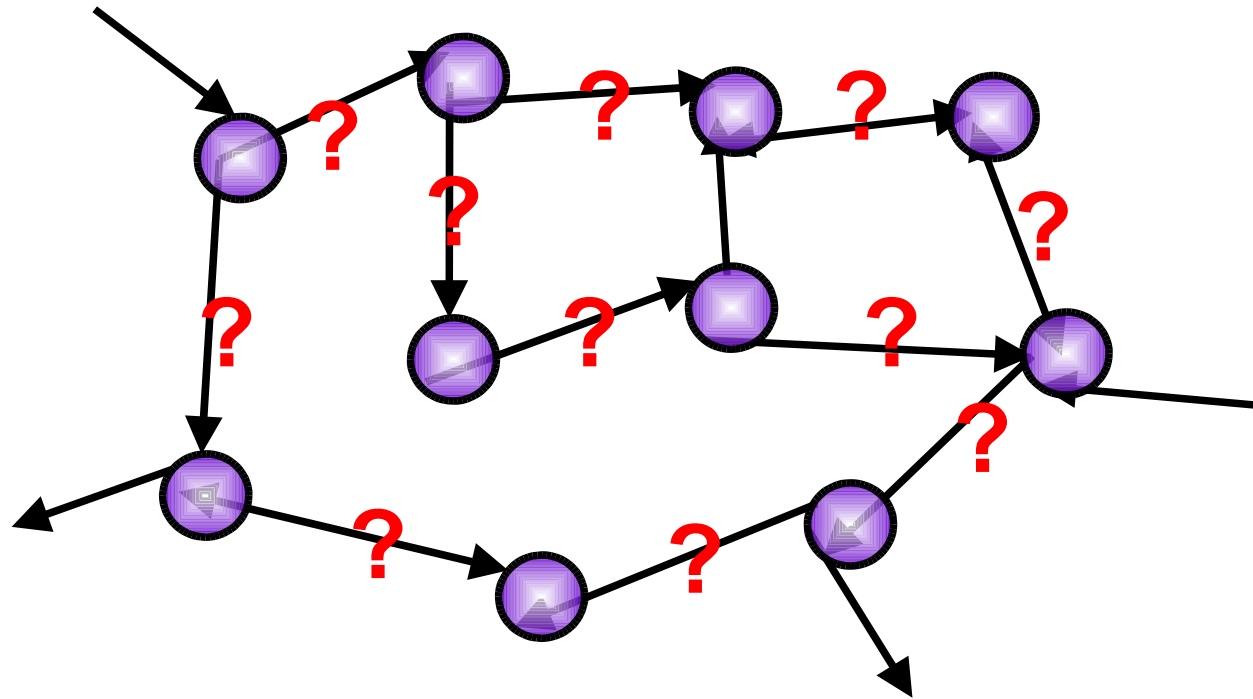


“Inversion of Control is about software Components doing what they are told, when they are told. Your OO application could well become unmaintainable without it.”

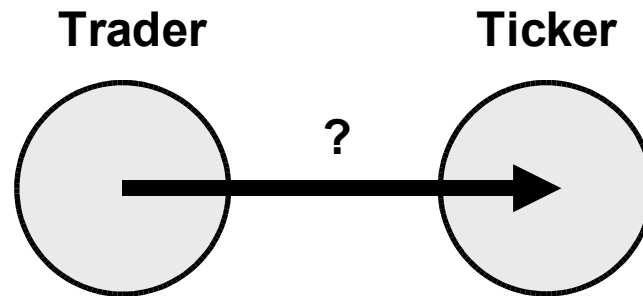
Paul Hammant, ThoughtWorks

- IoC – “The Hollywood Principle”
 - Dependency Injection
 - Dependency Inversion (DIP)
 - Lifecycle

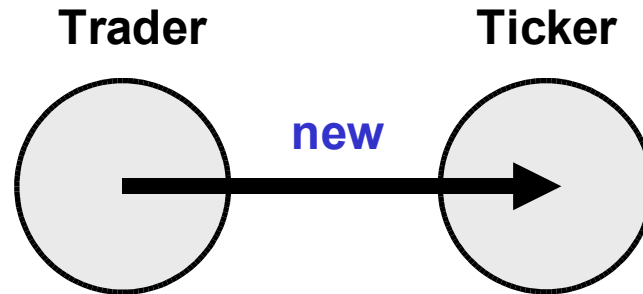
- How do these relationships get established?



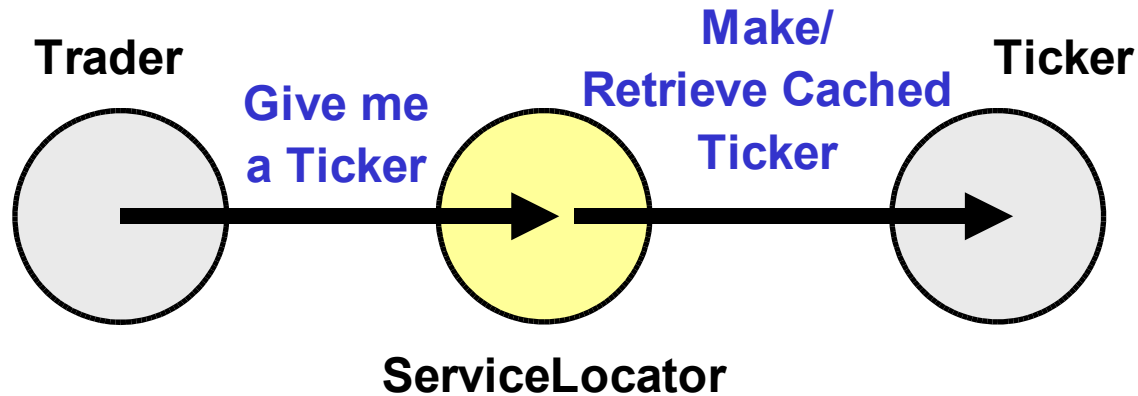
A simple example



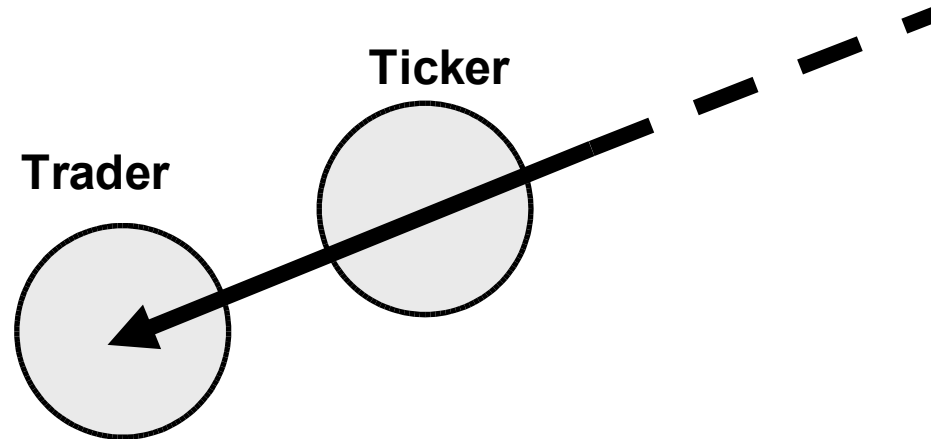
- A Stock Trader that buys stocks.
- Gets the price from a Stock Ticker.
- How does the Trader get wired to the Ticker?



```
public class Trader {  
    private Ticker ticker = new Ticker();  
}
```



```
public class Trader {  
    private Ticker ticker = (Ticker)  
        ServiceLocator.getInstance().  
        get(Ticker.class);  
}
```

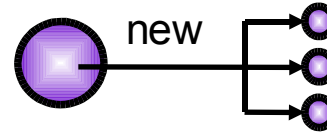



```
public class Trader {  
    private final Ticker ticker;  
    public Trader(Ticker ticker) {  
        this.ticker = ticker;  
    }  
}
```

Summary

- Explicit instantiation

- Components instantiate their own dependencies
- Kitchen sink problem
- Inhibits unit testing



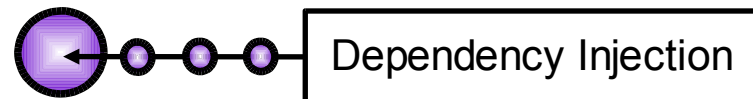
- Service Locator

- Components look up dependencies in a well known place
- Invasive
- Inhibits unit testing

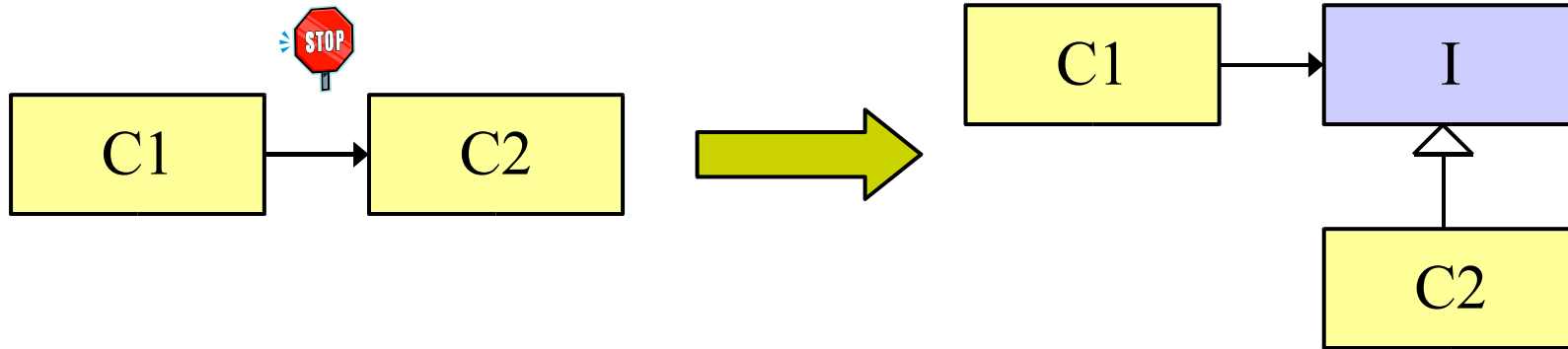


- Dependency Injection

- Components don't reach out to retrieve dependencies
- Instead they are handed dependencies by an external entity
- Non-Invasive



Dependency Inversion



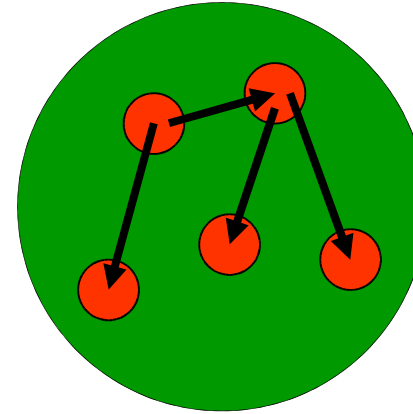
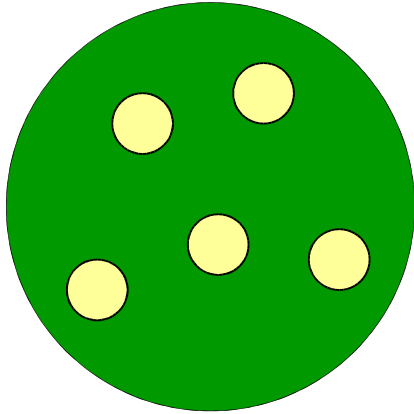
- Favours loose coupling → testability in isolation
- Components should be split in two parts
 - **Service**, a declaration of offered functionality (Java interface)
 - **Implementation**, a specific implementation of a service (class)
- Makes multiple runtime coupling combinations easy
- Breaks the dreaded “everything depends on everything” problem



“I was expecting a paradigm shift
and all I got was a lousy constructor”

Zohar Melamed, BNP Paribas

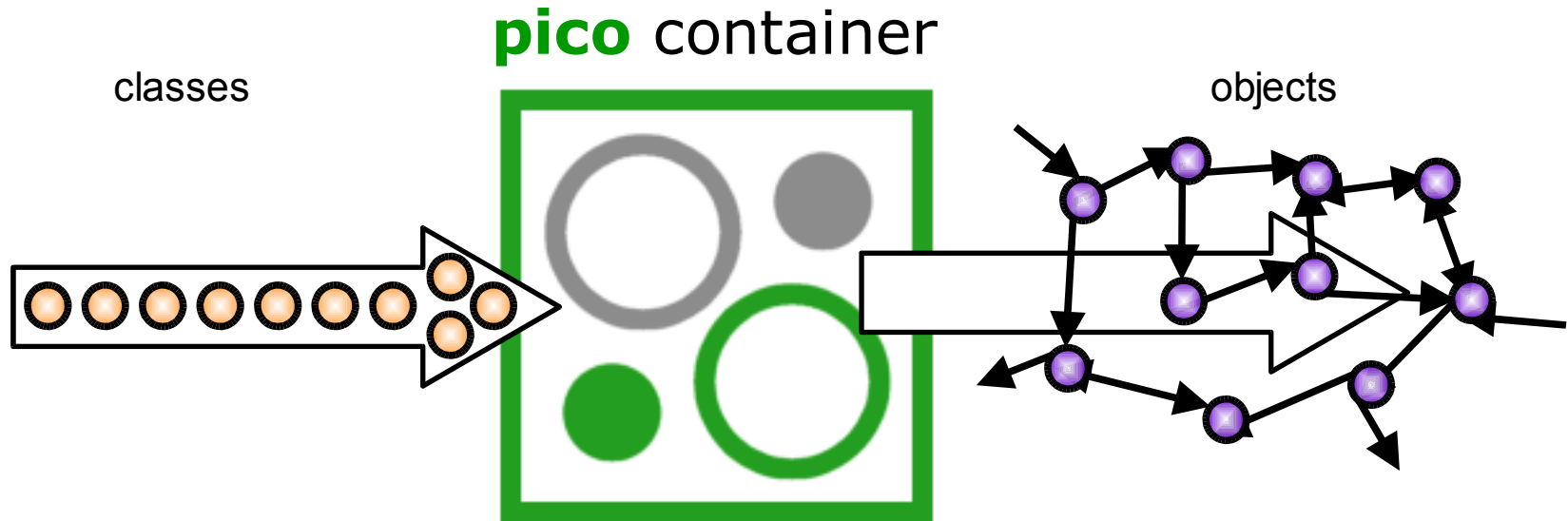
What does PicoContainer do?



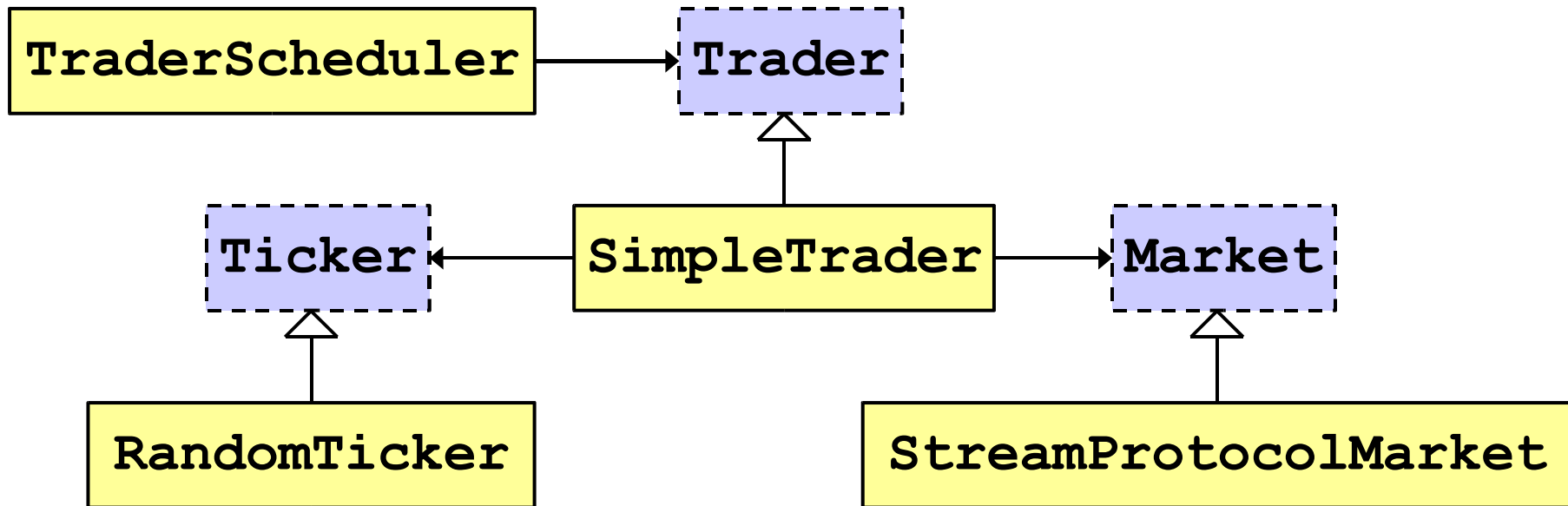
1. Register components

2. Materialize and lace the components

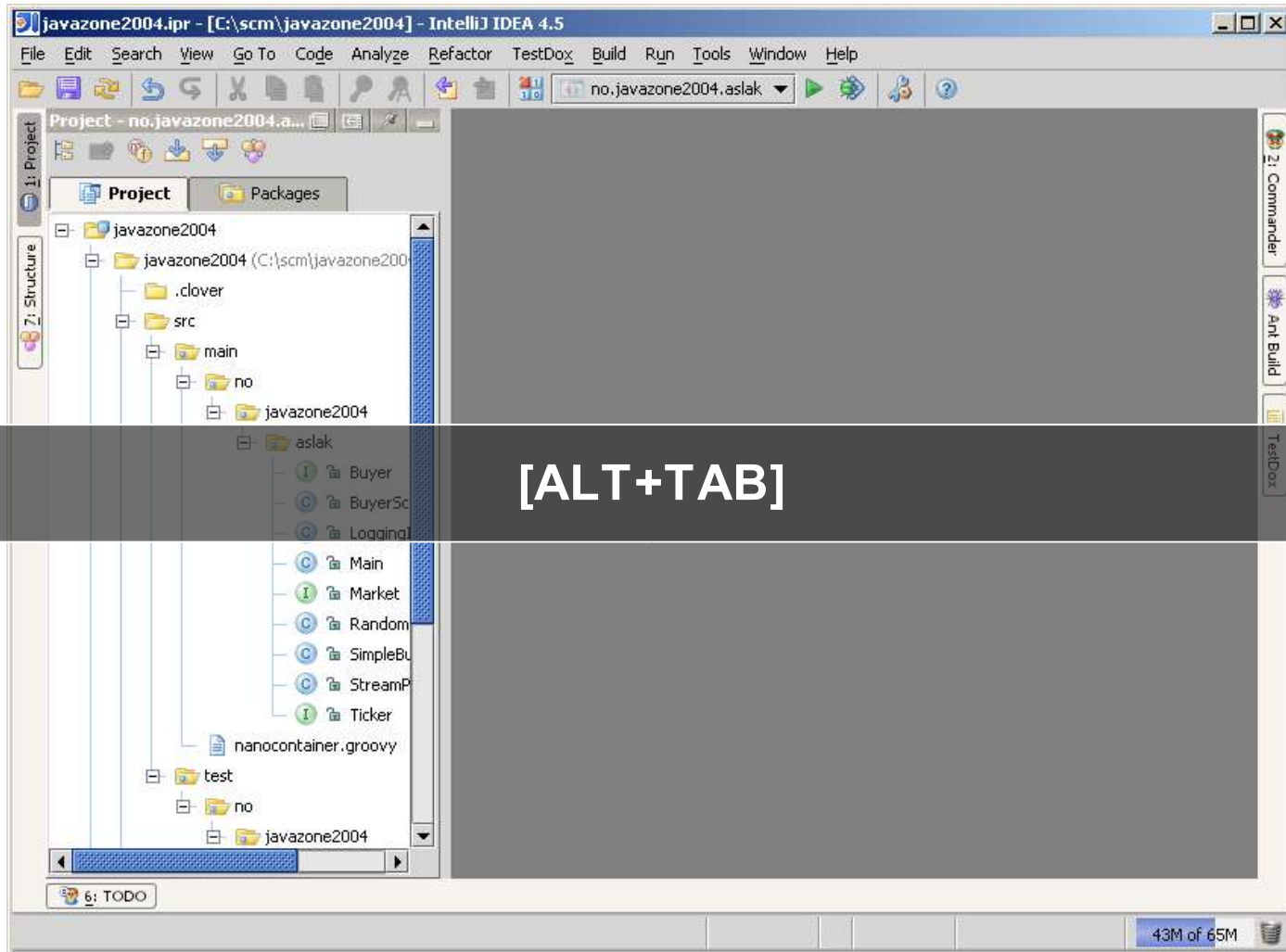
- A simple Dependency Injection container
- It works like a hash table on mescaline
 - Object instantiation
 - Dependency injection
 - Pluggable lifecycle management
 - Dependency hierarchies
 - Extensible via API and NanoContainer



Our toy app

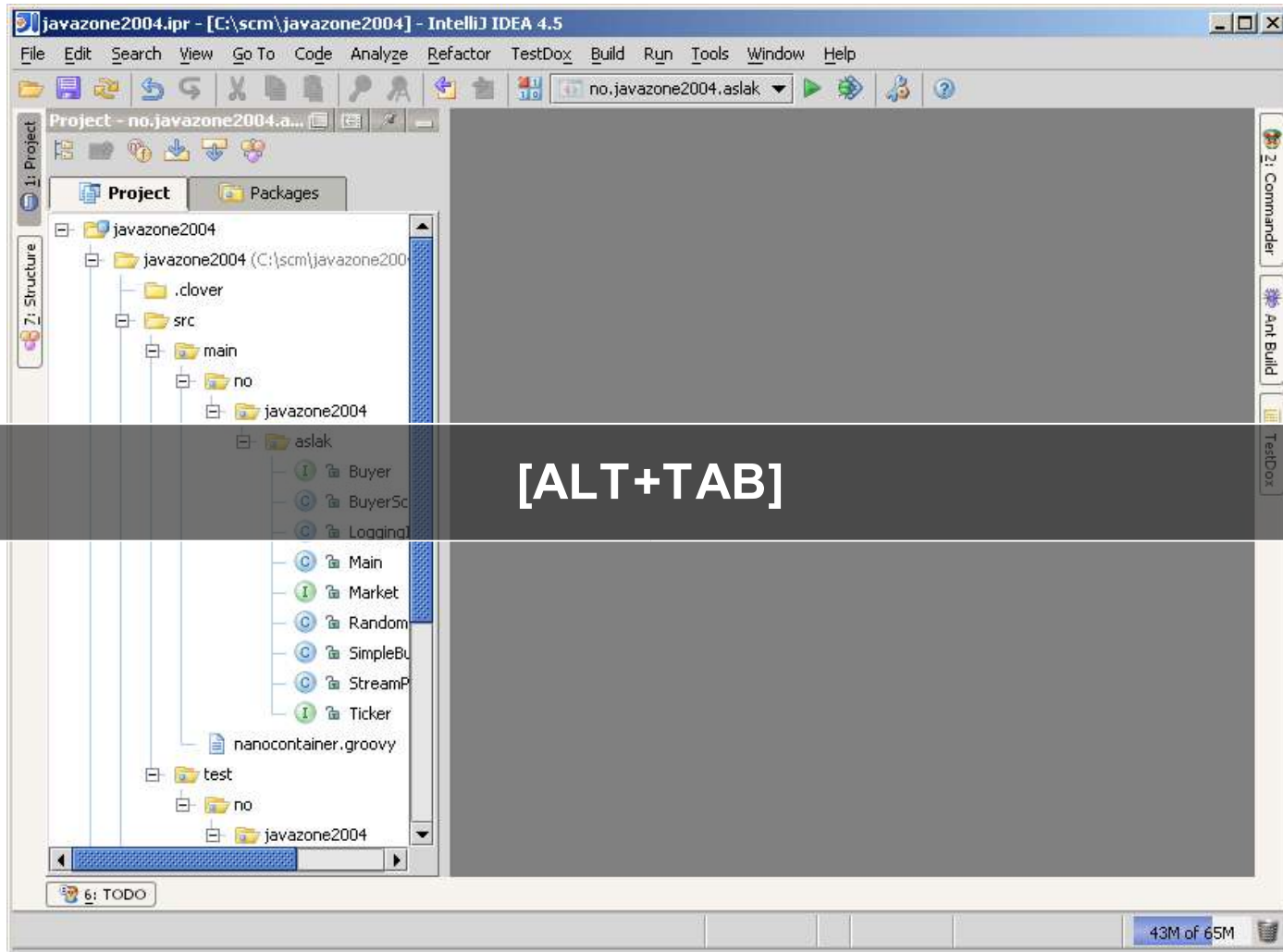


PicoContainer in action!



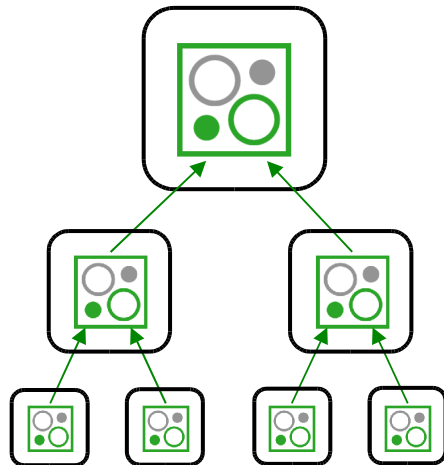
- Script front-end for PicoContainer
- Allows soft assembly of components in:
 - Groovy
 - Beanshell
 - Javascript (Rhino)
 - Jython
 - XML
- Adds powerful AOP capability (aopalliance/dynaop)
- Various other extensions
 - WebWork 1&2
 - Struts
 - Hibernate
 - JMX
 - And much more...

NanoContainer in action!



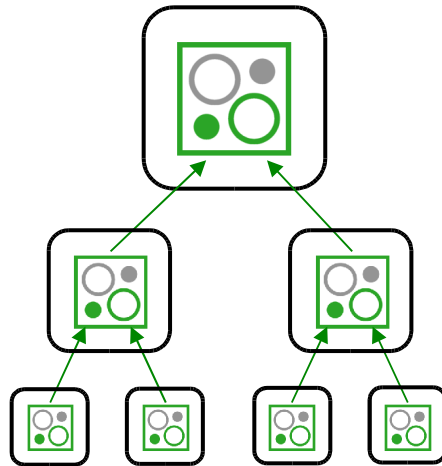
[ALT+TAB]

- Dependency Injection Pattern
 - Internal decoupling
 - Testability
 - Configurability
 - No-bullshit components
- PicoContainer
 - Low level Java API
 - Dependency Injection
 - Component Lifecycle
 - Non-invasive
- NanoContainer
 - Several high level script APIs
 - AOP
 - Integration with other frameworks



- No more static singletons
- Components in parent visible
- Lifecycle propagated down
- Core of complex assemblies
- Parallel to classloaders

- Brings DI and AOP to your web framework
 - WebWork 1&2
 - Struts
 - NanoWeb (our own framework based on Groovy/Velocity)



Web app scope
(`javax.servlet.ServletContext`)

Session scope
(`javax.servlet.http.HttpSession`)

Request scope
(`javax.servlet.HttpServletRequest`)

```
<web-app>
  <context-param>
    <param-name>nanocontainer.groovy</param-name>
    <param-value><![CDATA[
      ... Your groovy assembly script here (!) ...
    ]]></param-value>
  </context-param>
</web-app>
```

Thank You!

aslak@thoughtworks.com

<http://blogs.codehaus.org/people/~rinkrank/>

<http://www.picocontainer.org/>

<http://www.nanocontainer.org/>