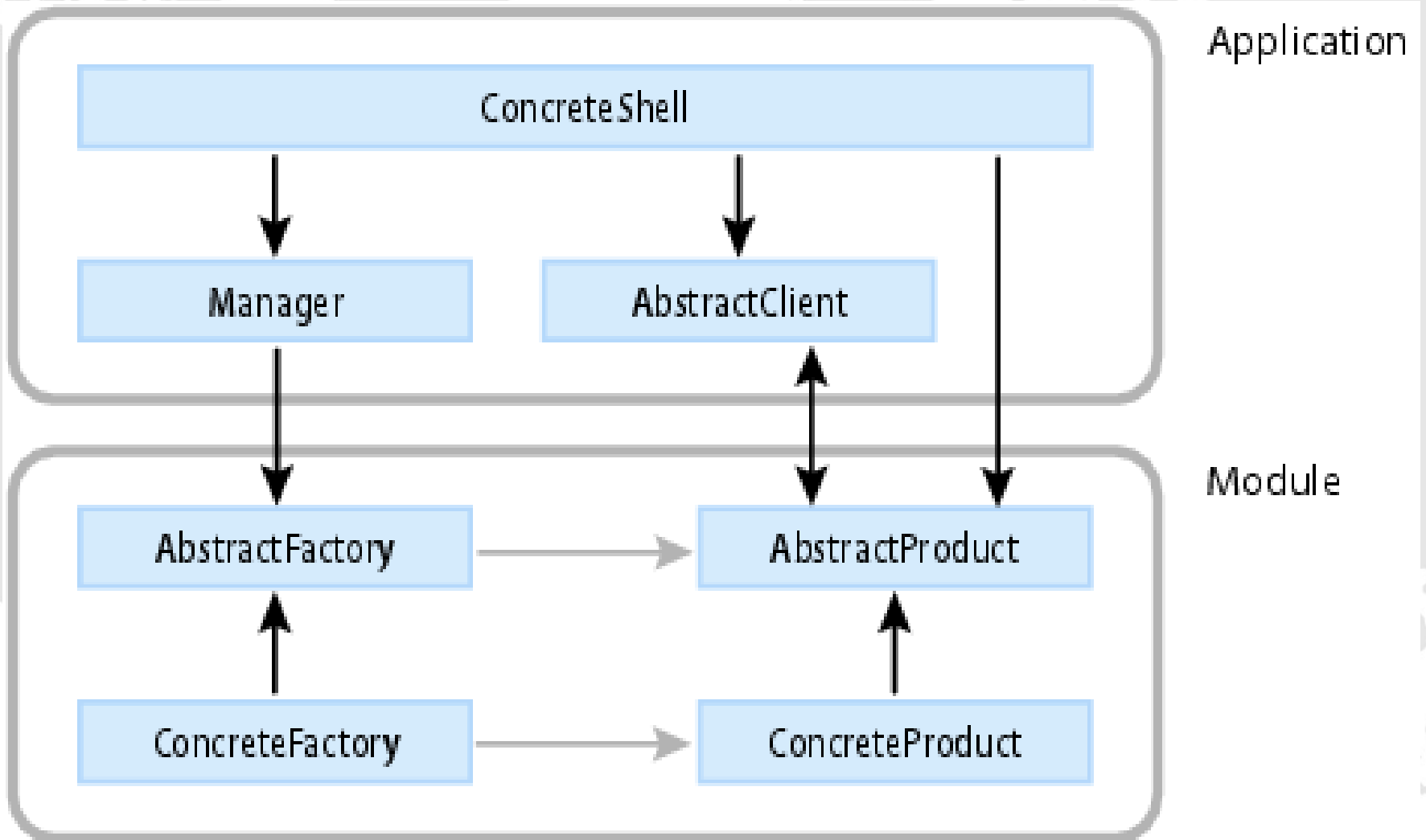


Използване на Flex Modules и RSLs

Бранимир Ангелов Ангелов

Мотивация за използването на Flex Modules

- Намаляване на времето за стартиране.
- По-лесна поддръжка.
- По-бързо и оптимално компилиране на отделните части на приложението.
- Възможност за динамичен избор на функционалност.





- ModuleManager
 - `getModuleInfo() : IModuleInfo`
- IFlexModuleFactory
 - `create()`
- ModuleLoader
- Module – MXML базирани модули
- ModuleBase – Actionscript базирани модули



- New MXML Module
- New Actionscript class базиран на ModuleBase
- Компилиране
 - Чрез Flex Builder
 - Properties --> Flex Modules
 - Optimize for Application
 - Чрез mxmhc



- MXML подход
 - ModuleLoader
 - loadModule() / unloadModule() методи
 - child – Вече заредения и построен модул
 - applicationDomain – домейнът в който искаме да се зареди дадения модул
 - events

```
<mx:ViewStack>
    ...
    <mx:ModuleLoader id="moduleLoader" url="MyModule.swf" />
    ...
</mx:ViewStack>
```



- Actionscript подход
 - `ModuleManager.getModule(url)`
 - `IModuleInfo.load(applicationDomain)`
 - `IModuleInfo.factory.create()`
 - `IModuleInfo.addEventListener()`

- Общ код или assets между различните модули.
 - Използвайте отделен модул в който пакетирате този код или asset-и.
 - Заредете него преди целевите модули в **application domain**, който ще е достъпен за целевите модули.

- Подобни на модулите, но се зареждат по време на стартиране на приложението
- Подходящи за споделяне на код или asset между различни приложения

- Unsigned RSL
 - Съхраняват се в browser cache-а
- Signed RSL
 - Ново в Flex 3
 - Изискват Flash Player 9.0.60
 - Съхраняват се също и във flash player cache-а