# What is the “base” module / cep-base.jar?

(Ashwin Jayaprakash, June 2010)

The base module is an attempt to provide some simple architectural components to build new BE components:

* Provide some very basic boilerplate code so that you do not have to re-think/redesign/code
* Build quickly
* Enforce some fundamental OOAD concepts
* Component oriented development
* Self-documenting code. (Seriously. Have a look at the annotations)
* Because I wanted to re-use the framework code that I wrote for Patterns
* cep-kernel is too Rule focused. cep-common is bloated. All the good names were already taken

cep-loadbal

/non-be

cep-loadbal/be

cep-base/be

cep-common

cep-kernel

cep-base

/non-be

rule-engine

cep-query

cep-pattern/be

cep-pattern

/non-be

# Why should I use it?

* You don’t have to
* If you are starting out on a new non-UI module, then it would be good to leverage this work

# What are the benefits?

* Currently, Pattern and LoadBalancer modules have 2 sub-modules each
  + Core
  + BE
* Core is plain Java. No dependency on BE. None at all
  + If you plan on grafting this to another product (AS) this is the ideal approach
  + Forces you to write code in separate and clean modules
  + Unit test is easy
* BE
  + This provides the integration layer with BE. This layer uses the Core layer

# Who is using it?

* As of now, Pattern and LoadBalancer

# What components does Base provide?

* A few simple ones. For now. See diagram below

Resource

ResourceProvider

JDK Logger

Service

Id

BE log adapter

AsyncExecutor

AsyncScheduler

IdGenerator

<Annotations>

Optional

ReadOnly

ThreadSafe

LifecycleException

LogCategory

<Other components>