

# AvalonRoadmap

## Avalon 5 (Spearhead) Roadmap

Here's an effort to define Avalon 5 (once codenamed Spearhead). Please join in!

The **process**:

- Assembly a list of *concerns* and *goals* that both the developer and user communities wish to see addressed in the forthcoming year. This helps us
  - Make sure nothing falls through the cracks
  - Identify how are users are using Avalon and what they want out of it
  - Identify developer resources – who wants to do what
- Structure these concerns and goals into a general roadmap of specific **milestones** for the next year or so
- Map these milestones into **JIRA** which we will use to track our progress
- If necessary, the PMC will exploring updating our Charter to reflect the goals of the community

Please note that the majority of discussion surrounding this material should happen on the mailing lists.

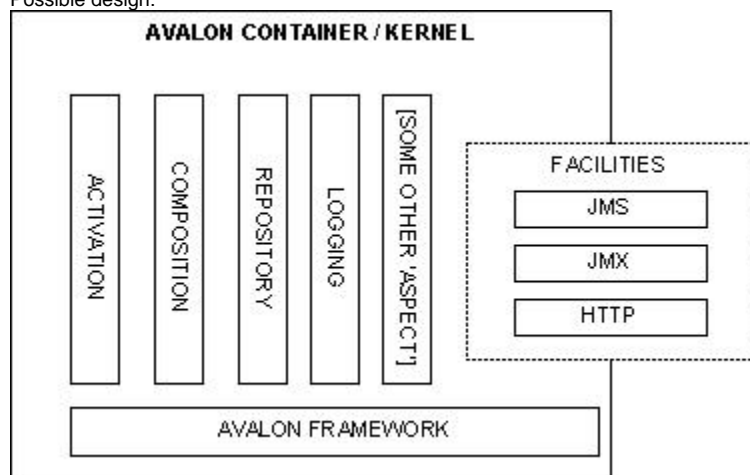
## Concerns and Goals

Okay, here's your chance to be heard! What's important to **you**? Where do you see Avalon in a year or two from now? What itches do you want scratched? *These are in **no** particular order!*

Note: this initial list was gleaned from thoughts and concerns brought up in Jan and Feb 2004 on the developers mailing list.

- "Put Avalon4 into maintenance and get on with Avalon5"
- Create an Avalon TCK, or set of TCK's including specs and tools (only worry about A5 for now?)
- One scalable container solution per language. ie- a real reference implementation
- Define Fortress as the Avalon 4 reference implementation. Promote Merlin as the eventual Avalon 5 reference implementation.
- Define an Avalon security model (include support for JAAS?)
- automated component publishing
- Container facilities: http, jmx, jms, iiop, rmi (altrmi)
- Support various IoC "types" (isn't that Avalon PI?)
- Cross container interoperability ( A5 should support A4 via adaptors. )
- publisher federation technologies (*uh, what is this?* 😊)
- Single meta format (between A4 and A5?)
- Service [AvailabilityContract](#) in the framework
- Circular dependency handling
- JCA client for/in Avalon
- Full Tutorial
- Update of "Developing With Avalon"
- Cooperation with the Repository/Depot folks and our own repository code
- Better (re)use and cooperation with Jakarta Commons – stop reinventing code, prefer Commons solutions to home grown when possible
- Make A5 "simpler" – less axioms to learn, more intuitive
- Unit Tests!
- IDE plugins (like Merlin Developer)
- "Codenames"
  - Merlin = Avalon 5 container (should support A4 – ECM/Fortress)
  - Fortress = the Avalon 4 reference implementation
  - Spearhead = ??? (enhancements, adjustments to the actual framework?)
- For Avalon 5 nothing should be sacred. Deprecate or actually drop marker interfaces, etc.
- Better documentation of Merlin's decomposition – how it works together, what's the absolute smallest container you could build from it
- An Avalon "kitchen sink" distribution – a full fledged Avalon app server like [EnterpriseObjectBroker](#)
- Cooperation and discussion with the larger IoC Community – get ideas and discussions with JContainer people, Pico, Spring, etc.
- Component libraries – what to do with Cornertone and Excalibur? What stays in Avalon, what goes to Commons or elsewhere. Will Avalon host a Component library/registry?
- Migration tools and documentation from A4 to A5
- A JSR?
- ... *add more by editing this page*

Possible design:



## Roadmap Milestones

A preliminary list of milestones.

[fixme by editing this page!!!]

---

*note: the old material is [here](#)*