

# YAHOO!

Code structure and Dependence  
Cleanup needed

Jason Kenny

## Why this is an issue

- Modifying code harder is harder to do:
    - Dependencies between component are unclear
    - Code is not separated in to different modules as it should be causing dependency issues
    - API use for a given task is unclear
  - Running and creating tests are harder than it should be
  - Adding or removing code is scary as we don't know it is "safe".
  - Black boxing of code ("it's magic, I can't touch it")
  - Reduces overall velocity of changes and quality of work.
- This will only get worse unless we address these issues!

## Some Issues (High level)

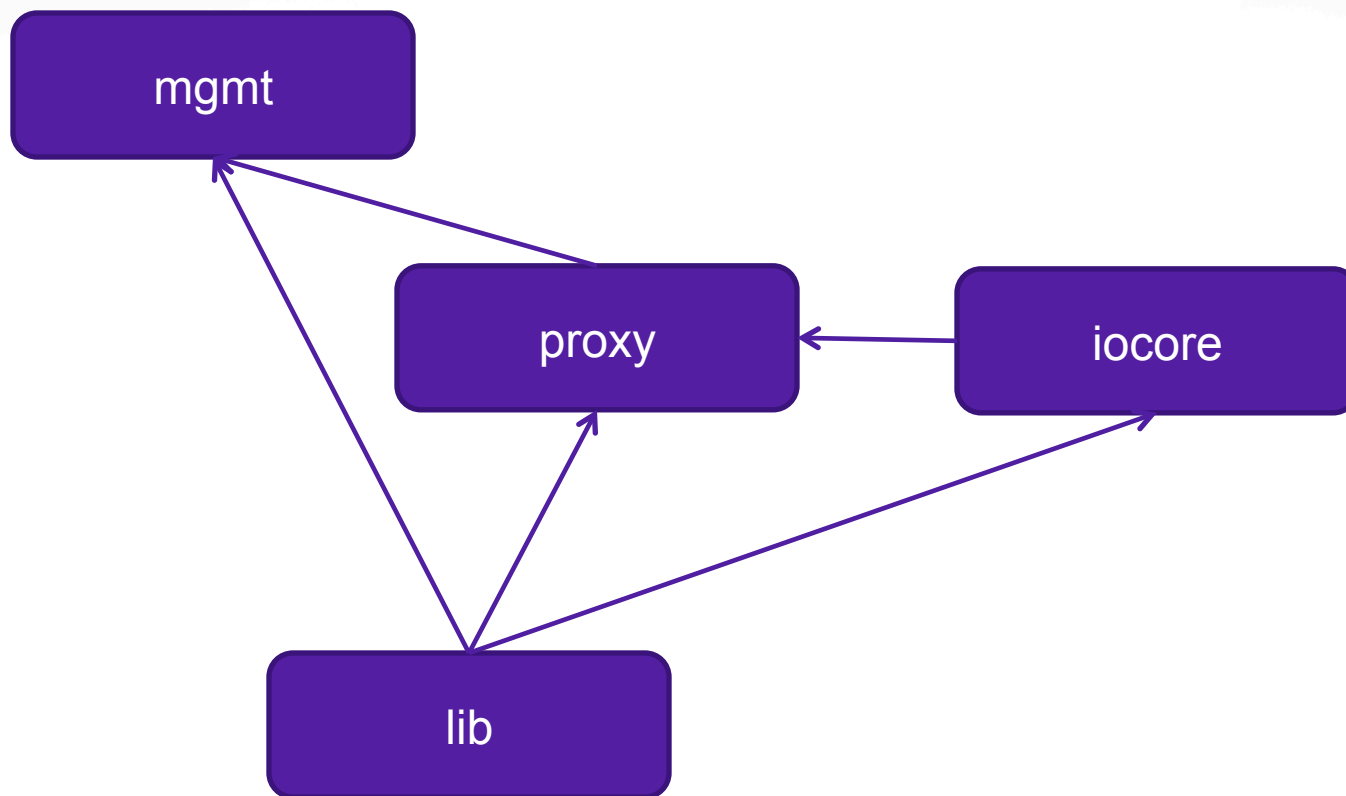
- There current code structure is getting chaotic.
  - Unclear architecture and design.
- Current Autoconf/make system does not support some basic features that it should support
- Lack of separation of test code from production code.
- Some items in build should be runtime values, not compile time values

## Issues- code structure

- A number of components for circular dependencies
  - Forces build environment requires in general all paths to all location
  - Lack of build component hierarchy in build.
- It is not clear why some component are where there are.
  - Example: proxy/hdrs is core code needed by low level components it seems like it should be in /iocore
- It is not clear where people should add new code

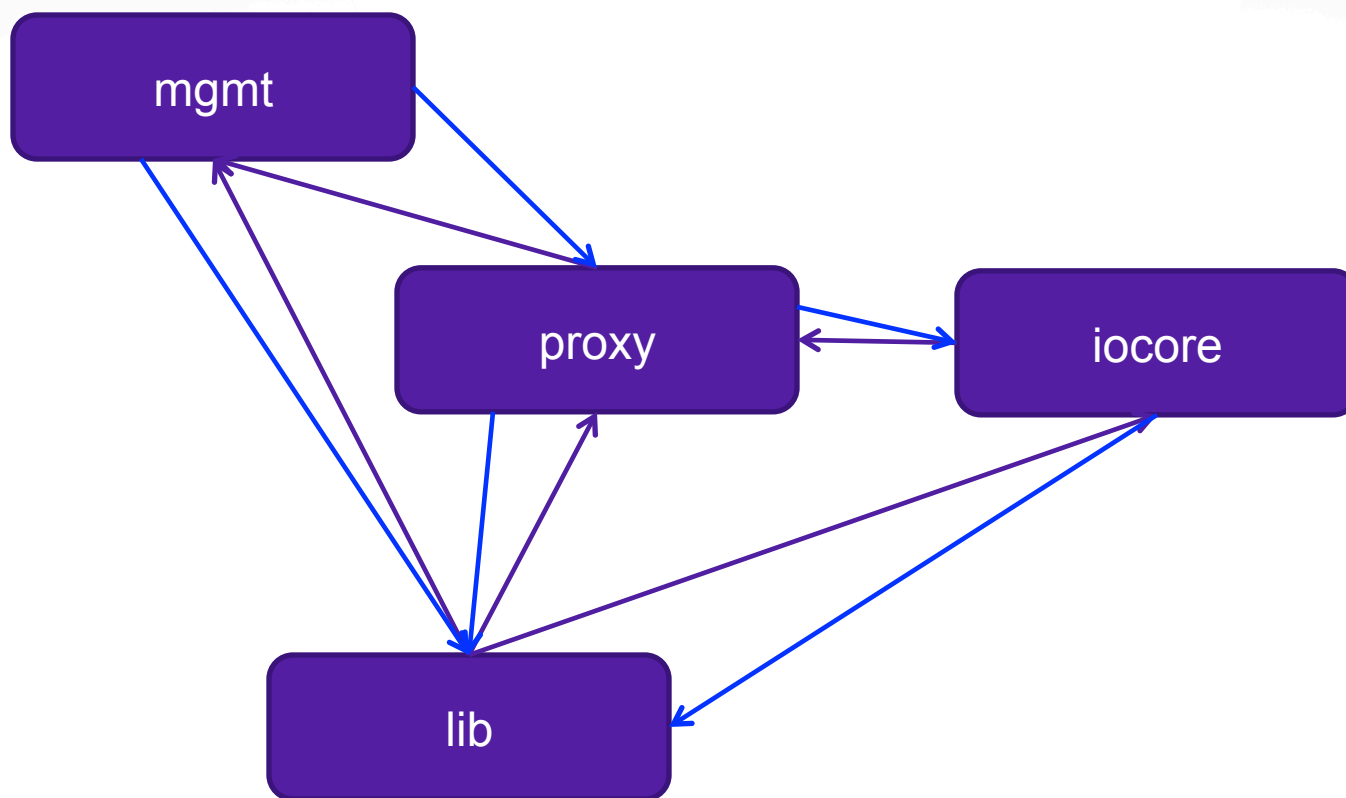
# Basic Depends

What it should look like

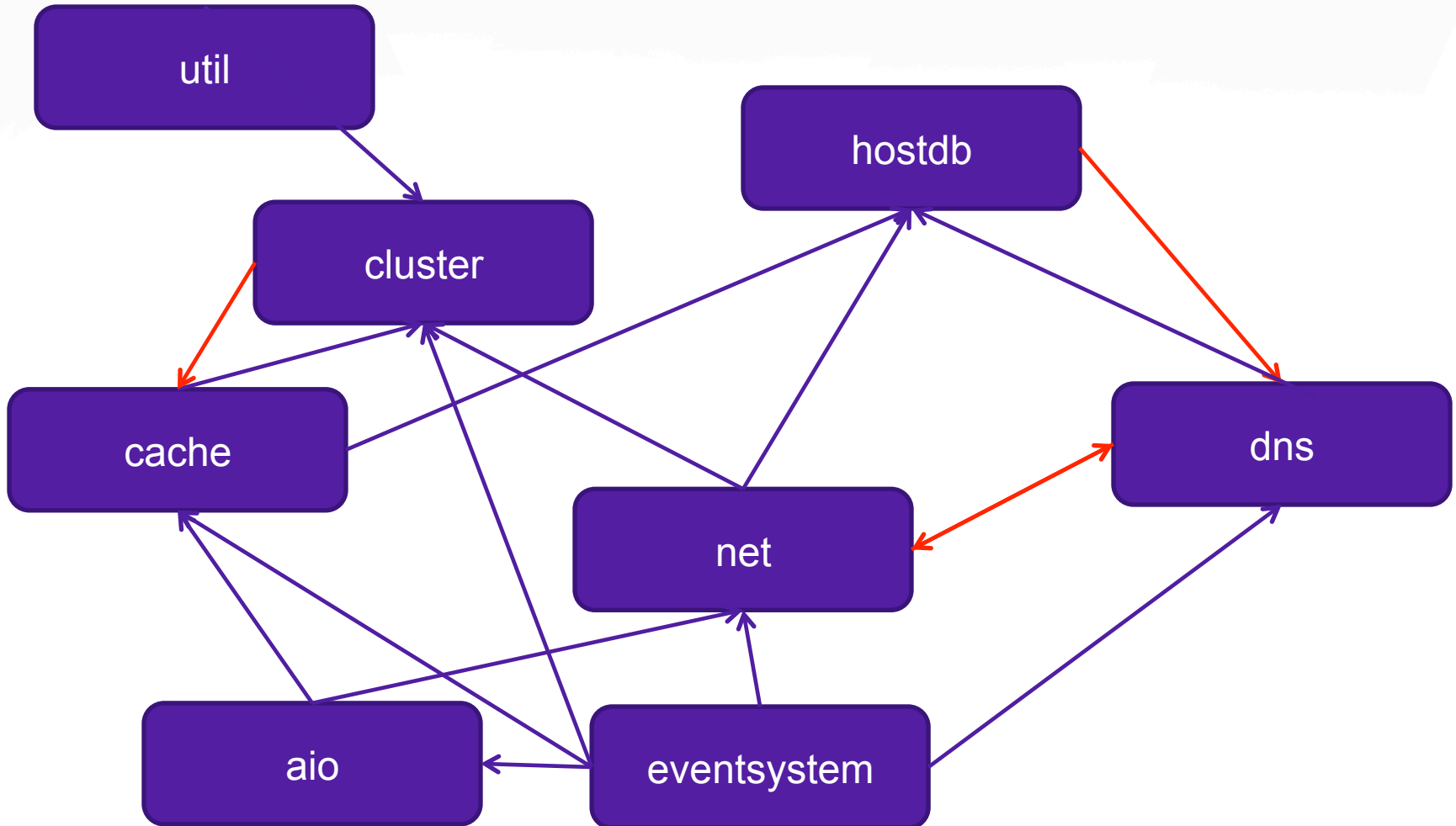


# Basic Depends

## What it really looks like



## Other dependency issues: iocore (there are more)



## Issues- Autoconf/make support

Love it or hate it... this is just focus on Autoconf/make should be doing correctly.

- DEST\_DIR feature does not work
  - this is complicated by some build time issues that hard code paths
- Out of source builds don't work
- -J based build are flaky



# Issues-separation of test code

- It is unclear what test we have
- It unclear where the test are
- Some tests are in main source files and some are separated in to different source modules
- Test code often can only run if the build is installed.
- Cannot be sandboxed easily.
  - Hardcoded paths make this harder solve.
  - Certain layouts just fail...

# What we need to do

- Make a simpler and more scalable code structure
- Fix up the build scripts
- Separate the tests code and group the test based on the type ( Unit testing, regression, etc..)
- Refactor the code:
  - Break up circular dependencies
  - Clean up include structure
  - Break up Uber-source files in separate files
  - Move items, such as configuration files or some runtime behaviors, to a runtime data driven model.
  - Use namespaces

Questions?