C Runtime In Tcl

Andreas Kupries

23th Annual Tcl Conference
Crowne Plaza River Oaks, Houston, TX
Nov 14 – 18, 2016

CriTcl – Introduction I

- Invented by Jean-Claude Wippler
- Tcl Operation & Use → 2 Levels
  - Script – Glue, Policy
  - System – Speed, Hiding, Bindings to externals
- C extensions – Easy to use, complex to write
  - Configure/autoconf, Make, Boilerplate C code, …
- C Runtime In Tcl – DSL using Tcl's strengths
  - Text processing, Templates, Glue to external cmds
  - Collects C code, arranges for compile on-demand (Tcl autoload facilities)

CriTcl – Introduction II

- Initial design geared towards dev system
  - C compiler needed at runtime
- Deployment to customers – Oops
  - Require C compiler on production system 😖
  - Rewrite to regular extension 😟
- Leverage Tcl again 😁
  - Read the input files as usual,
force an immediate compile & link cycle,
deploy the resulting library & package
  - ♫ You now have a build system

CriTcl – New Work

- Mainly keep the course
  - Lighten the burden of the developer
  - Reduce overhead
  - Automate
- Via
  - Add-on packages to handle specific common situations
  - Enhancements to the Critcl core

CriTcl – Core Enhancements

- Meta Data
- Stub Table Support
- Optional and Variadic Arguments (**cproc**)
- Custom Argument Types
- Capturing Output
- Code Location Management

CriTcl – Add-ons

- Generators
  - Ekeko's (Tcl_[Set|Get]AssocData)
  - String Pools – Custom Literal Tables
  - Enumerations
  - Mapping External Enumeration
  - Mapping External Bitsets / Flags
  - Classes & Objects
- Build utilities

CriTcl – Location

- We are @ **github**
  - https://andreas-kupries.github.com/critcl
  - https://github.com/andreas-kupries/critcl

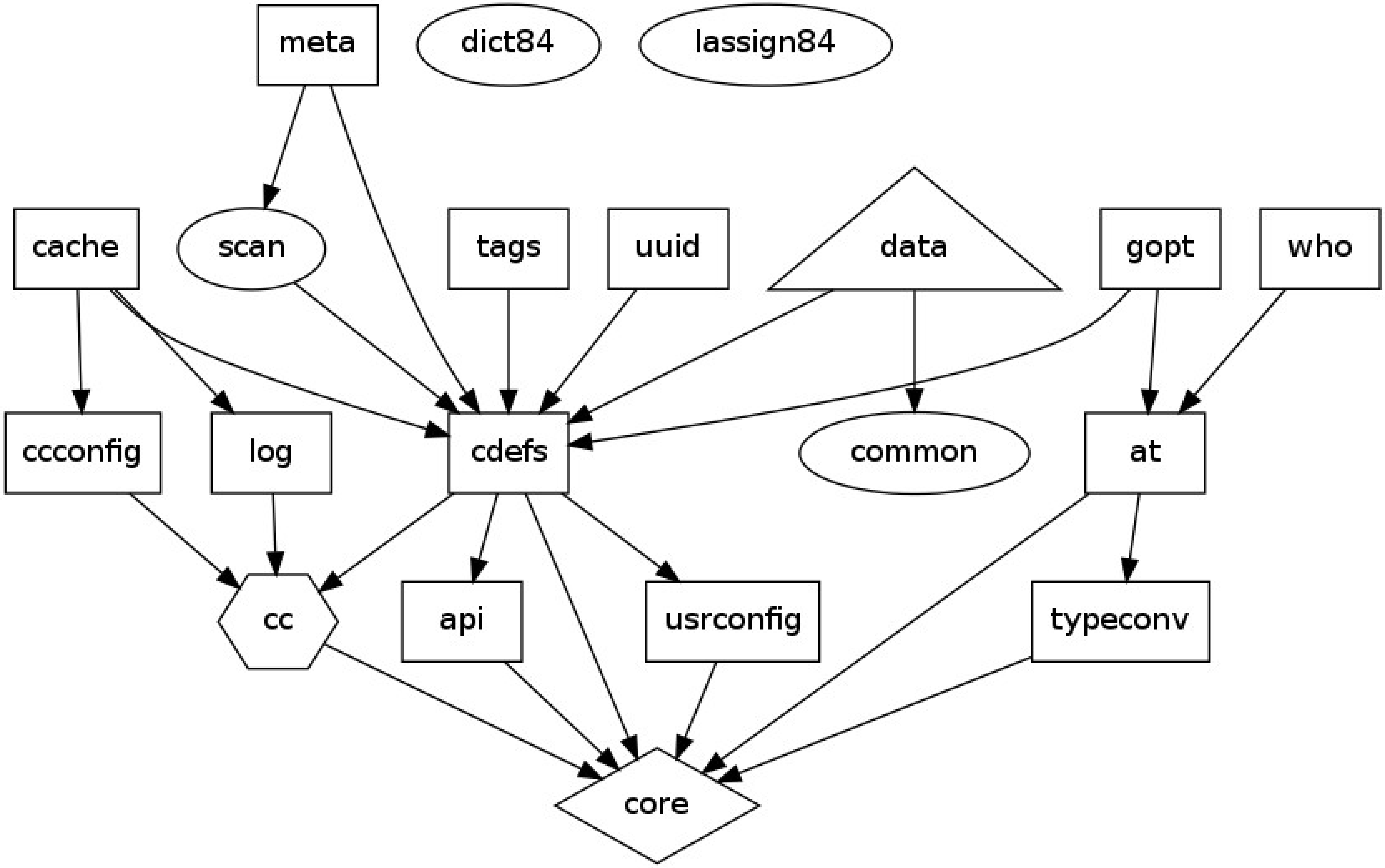
CriTcl – Ahead

- Possible DSL enhancements
  - Named parameters / options
  - Support for upvar
  - Parameterized types
    - list<T>, list<list<T>>
    - &T – See above, upvar
    - int[0..256], int [-1..1], …
    - …
- More packages
  - Useful C code fragments (Alloc, Trace, …)
  - ...

CriTcl – Ahead

- Slice & Dice branch
  - Slice 150K monolithic core into pieces
    - Easier to understand and maintain
    - Better internal APIs
    - Encapsulate CC access apart from general management
      - Add **tcc4tcl** support

CriTcl – Future Ahead


meta
dict84
lassign84
cache
scan
tags
uuid
data
gopt
who
ccconfig
log
cdefs
common
at
cc
api
usrconfig
typeconv
core