SAMBA

opening windows to a wider world

The Samba Tour of Scripting Languages

Andrew Bartlett and Amitay Isaacs

Samba is a C project

- Parts of the team have always had an aversion to scripting languages
- C is the only truly portable thing
- Except for all the exceptions...

Samba over the past two decades

Starting as you might expect...

- Build systems
  - Make, autoconf, but never automake
- Auto-generation
  - awk for auto-prototypes of C functions

M4, how do I loathe thee?

- Autoconf gone mad
  - Portable to multiple versions of autoconf templates
  - Including m4 files from subprojects

Every project needs a build system

- Samba4 started to develop a separate identity
- Shared libraries
- Grouping of code
- From object lists to subsystems
- Still autoconf, m4, and make but now also perl

Building from IDL

- An awkward way to build from IDL
  - An early attempt at an IDL compiler was written in awk
  - Sadly the results were hand-edited
  - Restricted by Samba's desire to be incredibly portable
- PIDL revived
  - Perl-based IDL compiler
  - Results used as-is, or exceptions made in the source IDL
  - All of Samba now uses PIDL extensively

Javascript before it was cool

- Before the days of node.js Samba had an embedded Javascript engine
- Based on EJS
- C bindings for RPC functions
- Provision script to lay out a template database

But the cool kids were using python

- Suffering from being ahead of our time for once
  - We ditched JS and moved to python
- Tridge had to be subtly mislead to accept it
  - (the promise of easier debugging)
  - but is now a big python fan..
- Writing code in an exception based language is much cleaner

Python bindings

- IDL generated bindings
  - Call any remote dce/rpc funtion
  - Build any IDL-based structure
- Also bindings for C interfaces:
  - ldb
  - tdb
  - talloc objects
  - Essentially all useful parts of Samba have or can get python bindings

Python scripts

- samba-tool gradually rewritten:
  - from all-C
  - to python
  - or python wrapped around C
- Forking python scripts from the main 'samba' process to handle small tasks

Testing framework

- Perl test framework
  - Creates test environments
  - Calls provision, starts server processes
- Python 'subunit' test result processing
- Tests written in:
  - C – smbtorture
  - Python
  - Shell

Revamping the build system

- It all started with a simple proposal: cmake
  - Our existing m4/autoconf/GNU make system was a failure
  - So the case for replacing it would be easy, right?
- Counter-proposal: waf
  - Written in python
  - Used to build Samba4 at first
- The eventual result:
  - Samba3 still built with m4, autoconf and make
  - A combined build using waf

Python: Samba's scripting language

- Slowly replacing other languages
- Perl will remain for PIDL
- Difficult to dislodge the m4, autoconf and make build system
-