

THE DECLINE AND FALL OF BIND 10

Shane Kerr <shane@time-travellers.org>

Open Source Working Group

RIPE 68, Warsaw

2014-05-14



PART I: THE STORY OF BIND 10

Prehistory



- BIND was developed in the mists of time
 - In many ways, BIND was DNS
 - Open source insured wide adoption
- BIND 4 begat BIND 8, which begat BIND 9
 - Each was born as a child of its time
 - Each had vision and design shaped of that time
- Then Vixie declared it time for a new BIND

An Open Source Fairy Tale....



Once upon a time, there was a little coder named Shane.

He worked at a magical company called ISC. But he wanted to do software engineering, and all of the other coders loved hacking and thought he was stinky. So one sad day went to a new company where he could make code that was well-designed and tested.

After a while, ISC called Shane and said, "would you come and make a BRAND NEW software called BIND 10?" He said "Yes!", and they all lived happily ever after...

The Honeymoon Phase

- Set basic parameters early on
 - *Community* open source project (public wiki, lists, repository)
 - Dual languages (Python & C++)
 - Minimize re-invention
 - High-standards for code quality, testability
- Team gathered
 - Existing BIND 9 coders, new coders
 - Non-staff developers added
- Code happens
 - Basic infrastructure, libraries
 - Toy DNS server developed

The First Release – the Good

- After 1st year, made first public release
- Met the project goals
 - Simple authoritative-only server
 - Underlying infrastructure more-or-less working
 - DNS library working
- On time
- Under budget

The First Release – the Bad

- Lots of technical debt
 - Natural (unavoidable) from deadline-driven dev
- Sky high expectations
 - “That seemed easy, now implement the rest.”
- Sponsors unhappy with going under budget
 - Really.
- Nothing usable for production
 - “It works, but...”

The First Release – the Ugly

THIS PAGE INTENTIONALLY LEFT BLANK

Year 2: End of the Beginning

- Ambitious Goal: recursive resolver
 - Analogy: DNS authoritative as web server, DNS recursive as web browser (Firefox source is 23x bigger than Apache, so maybe not so bad...)
- Coding resources (a.k.a. “budget”)
 - Original plan – double budget
 - Actual sponsorship – extra 10-15%
- 2-3 months to pay back technical debt
- Lack of expertise
 - Team of DNS and software experts
 - Nobody that had actually built a DNS resolver

Year 2: Beginning of the End

- Change of Plans
 - One sponsor wanted production-ready authoritative
 - By end of Q3 (remember 2-3 months behind!)
- Split Team
 - R-Team continue recursive work
 - A-Team now works on authoritative server
- Results Predictable
 - Recursive server a toy, no DNSSEC
 - Authoritative not production ready
 - Sponsors unhappy at change of plans and lack of success

To Year 3... and Beyond!

- *Lots* of technical debt
 - Authoritative server not yet production ready
 - Recursive server pretty weak
 - Tons of underlying work to do
- Understandably few real users
- Sponsors increasingly restive
 - More on the joy of sponsored development later...
- Internal ISC support fails to materialize
 - New BIND 9 manager applies lipstick to pig
 - Business can't work with a product without users
- NSD progresses, Knot & Yadifa appear, PowerDNSSEC released

IMPERIAL SUCCESSION AT ISC

- Paul Vixie steps down
- Barry Greene steps up
- Barry fired by the board
 - More than 3 months without leader
- Kannan Ayyar hired by the board
- Madness ensues*
- Kannan fired by the board
- Jeff Osborn hired by the board



* Really. It was bad.

THE SACKING OF BIND 10

- Several sponsors withdrew
- Team members started leaving
 - No permission to replace
- Development put “on hold” for year 5
- After 50% of sponsors left, funded model ends
- ISC board shuts down project
 - Part of overall 25% staff cuts



s / BIND ? 10 / Bundy / gi ;

- ISC letting BIND 10 loose into the wild
- Talked with interested parties to get friendly fork
- Bundy project spooled up
 - Hosted on GitHub
 - Web site, mailing lists, IRC, and so on...
 - A bit of discussion, some bugs & patches arriving
- Informal BoF-like gathering at RIPE 68
 - @Ujazdow room, 18:00 Thursday

PART II: OBSERVATIONS & RECOMMENDATIONS

Go Big or Go Home

- Managing transitions is hard
 - New system has less functionality, is different
- Building BIND 9 and BIND 10 was a bad idea
 - Expensive
 - Unclear message to community
 - Confusion about priorities internally
 - Impossible to reach feature parity when making old software better!
- Recommendations:
 - Only develop one codebase
 - Devise an evolutionary plan for replacement

Convince Your Suits

- BIND 10 originally ran in “skunkworks” mode
 - Separate team
 - New practices, procedures, tools, technologies
- A transition to mainstream mode is critical!
- Business folks will resist new software
 - Customers want what they have now... better
 - Don't care about technology
- Recommendation: therapy session



“This is a large customer resolver running BIND 9...”

No, please...

“And this is the startup time for a large zone...”

NOOOOOO!!!! AAAAAAAAAAAAAA!!!!

Sponsorship

No servant can serve two masters: for either he will hate the one, and love the other; or else he will hold to the one, and despise the other.

Luke 16:13

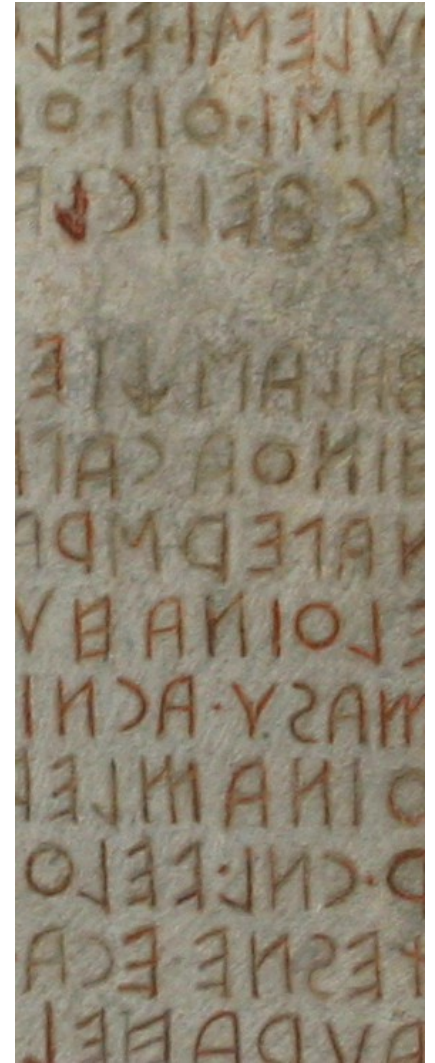
- BIND 10 had more than 10 sponsors
 - Sponsors mostly sat on a *Steering Committee*
 - Each company has unique goals
 - Each company has unique expectation
 - Frankly, it was a nightmare :(
- Recommendations
 - Sponsors are great! Try to get those!
 - Don't let any one sponsor dictate project direction
 - Either handle donations like a contract, or completely without strings
 - Make visible costs of requests for more reports, meetings, and so on

Some Maxims (with Caveats)

- “Release Early, Release Often”
 - Of course...
 - But you need *real users running the code*
- “Eat your own dog-food”
 - Of course...
 - But you need the mission-critical, customer-facing parts of your production systems to be force-fed dog-food over the screaming objections of your operations and business folks

LINGUA PYTHONIS

- Want a safe, fast, popular language
- Python is awesome, but too damn slow for DNS
 - Also Python is in the midst of a decade(s)-long effort to migrate from Python 2 to 3
- Popular compiled languages was basically C/C++
 - Maybe Go would be an acceptable option today
 - Maybe Rust in the near future
- People REALLY CARE what language software is written in
 - Makes sense for people who want to develop
 - Makes little sense otherwise, but...
 - Administrators really hate Python. Really. HATE.
- Perfect **bikeshed** topic
 - Someone will complain about any choice



To Re-Use, or Not to Re-Use (Code)

- BIND 10 chose to use libraries when possible
 - Well-designed, somewhat popular, maintained
 - Basically, had to pass a “sniff test”
 - BSD or BSD-friendly license (no GPL, MPL, ...)
- Makes installation a hassle
 - Worked to get libraries as packages in OS distributions
 - Actually not that hard!
 - Administrators hate dependencies. Rabidly. HATE.
- The other option is to not use libraries
 - Write the same functionality, not as well as a distraction
 - Or just leave the functionality out...

Provide Specific Value

- BIND 10 has a lot of interesting software ideas
 - Run-time configuration
 - Extensible design
 - Co-operating processes for robustness
 - And so on...
- DNS operators don't care about software
 - Care about speed, load time, memory footprint, feature set
- Suggestion: provide ***new*** value from day 0

People Fear & Hate Change

- BIND 10 is quite different for administrators
 - Lots of dependencies, slow build
 - Lots of processes
 - Tool to configure, not configuration files
- People hate change. Administrators especially.
- Suggestion: create a roadmap for changes
 - One at a time
 - Changes that work, great! Otherwise, back it out.
 - Don't exceed users' tolerance for new stuff.

