Upcoming plans for pkg 1.4

Vsevolod Stakhov
vsevolod@FreeBSD.org

FreeBSD

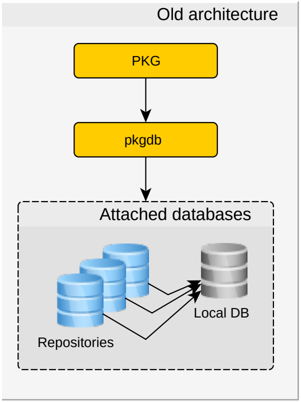
July 7, 2014

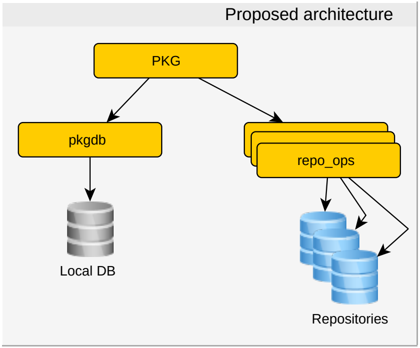
The list of main features

- Abstract repositories (ports, CPAN, CTAN...)
- Flexible dependencies
- Package popularity contest
- Packaging of the base system

Abstract repositories

Repositories architecture





Abstract repositories

Advantages of the new architecture.

- No longer limited to sqlite3 repos.
- Better multirepo support (without attach hacks)
- Allows to create repostitories of many types:
  - Ports
  - C($T|R|P$)AN
  - Base system?
  - . . .

Repository operations

Repository operations.

Operation	**Description**	**Mandatory**
Open	Opens repository	+
Init	Initialize repository structure	+
Create	Create repository	+
Close	Close repository	+
Update	Updates repository [1]	+

[1] this could be some cache building for some repo types

Repository operations

Packages operation

Operation	**Description**	**Mandatory**
Query	Search for a package or packages	+
Shlib	Get information about provided or required shared libraries	-
Ensure_loaded	Load specific fields of a package	+
Fetch	Fetch package from a repository[2]	+

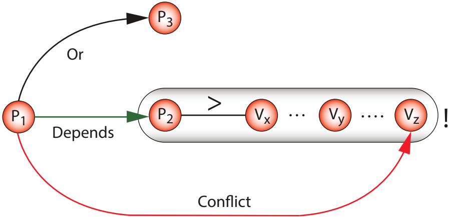
[2] or compile it

Flexible dependencies

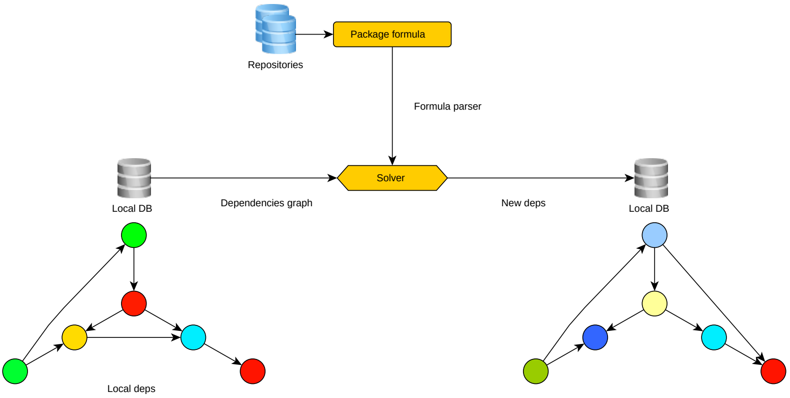
New dependencies format

$$libblah >= 1.0 + option_1, +option_2 || libfoo! = 1.1$$

- Can depend on normal packages and virtual packages (provides)
- Easy to define the concrete dependency versions
- Alternative dependencies



Flexible dependencies solving



Popularity contest

Purposes

- We want to gather statistics about packages using
- We can get statistics of OS versions/architectures
- We can check what options are mostly used
- We can estimate efficiency of security updates

Problems with surveys

1. We must protect all statistical data transferred from active and passive attacks
2. We must not ask for user's identity or establish it in any way
3. We should be tolerate to keys compromising and provide forward secrecy
4. We should protect server from malicious data and flooding
5. We need to traverse throught NATs and corporate firewalls

Solution proposed

- Use EDH over HTTP requests as key exchange procedure
- Store long-term public key inside pkg binary (should be used for offline signing)
- Identify a client by some random UUID
- Suggest to a client some crypto puzzle:
  - Relatively hard for unknown UUIDs
  - Simpler puzzle for subsequent connections
  - Very complex if submit rate from UUID is higher than normal

Packaging base

Problems arise

1. Too many build options available
2. Number of packages from base system is not defined
3. Need to export build options to system profile
4. Configuration update and mergemaster need special care

FreeBSD

Questions?

`vsevolod@FreeBSD.org`