Libvirt

What did we do wrong?

Michal Prívozník - Senior Software Engineer
Red Hat, Inc.
KVM Forum - 2015-08-25
What is Libvirt?

- C library, bindings
- Stable API
- Multiple hypervisors
- Host management
What is Libvirt?

Libvirt internals
What did we do wrong?

APIs missing some arguments

- `virDomainShutdown(virDomainPtr domain)`
- `virDomainShutdownFlags(virDomainPtr domain, unsigned int flags)`
What did we do wrong?

APIs missing some arguments

- virDomainCoreDump(virDomainPtr dom, const char *to, unsigned int flags)
- virDomainCoreDumpWithFormat(virDomainPtr dom, const char *to, unsigned int fmt, unsigned int flags)
What did we do wrong?

APIs too tied to a specific hypervisor

- `virDomainCreate(virDomainPtr dom)`
- `virDomainCreateWithFlags(virDomainPtr dom, unsigned int flags)`
- `virDomainCreateWithFiles(virDomainPtr dom, unsigned int nfiles, int *files, unsigned int flags)`
What did we do wrong?

APIs inherently racy

```c
int numActive, *idList, i;

numActive = virConnectNumOfDomains(conn);
idList = malloc(sizeof(int) * numActive);
virConnectListDomains(conn, idList, numActive);

for (i = 0; i < numActive; i++) {
    virDomainPtr dom = virDomainLookupByID(conn, idList[i]);
    /* Do something useful with @dom. */
}
```
What did we do wrong?

From bare C and glibc:

```c
free(NULL);
```

Not in libvirt:

```c
virDomainFree(NULL); /* Causes an error */
virDomainInterfaceFree(NULL); /* No error */
```
What did we do wrong?
Bad naming

- `virDomainDestroy()`
- `<controller type='pci' model='pcie-root-port'>
  <model name='ioh3420'/></controller>`
What did we do wrong?

XML elements

- `<xml/>`
What did we do wrong?

Internal code

- Introducing a new feature sometimes means rewriting a significant portion of directly not related code
What did we do right?

Internal APIs

- Majority of our API is internal
- For instance, many requests for turning virCommand into a separate library
What did we do right?

Internal objects

- Objects are not exposed to user
- Public header file:
  
  ```c
  typedef struct _virDomain virDomain;
  ```
- Private file: struct _virDomain { ... };

virConnect object has been adjusted more than 20 times.
Analysis
APIs missing some arguments

- Historical reasons
  - Libvirt started as a Xen wrapper
- Changing requirements
  - Hard to do proper analysis and design
Analysis
APIs too tied to a specific hypervisor

- Many one-man-show drivers
  - Lacking knowledge of approaches though all HVs
- Downstream deadlines
  - APIs pushed without driver implementation
Analysis

Traditional software vs Open source

- Hard to follow traditional SE models (e.g. waterfall model)
- OSS is able to go back and forth in SE models
- Perhaps some agile methodologies
- Daily SCRUM over IRC?
Analysis

Traditional software vs Open source

eXtreme Programming has some answers:

• Refactor
• Write tests (both unit and integration)
• Release often
Resolution

Traditional software vs Open source

- We did some mistakes, but we've learned from them
- Use common sense
- Be nice to new contributors
<Thank You!>

mprivozn@redhat.com

www mprivozn.fedorapeople.org