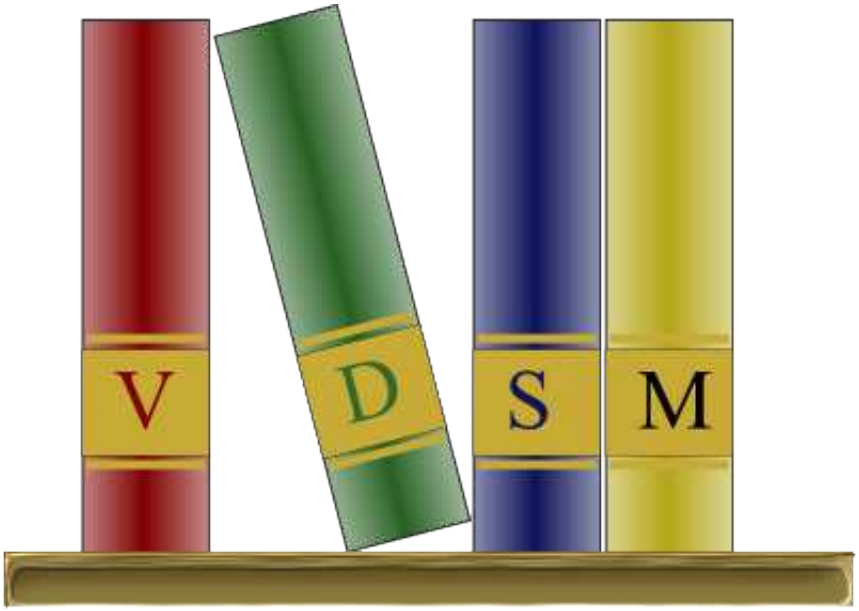
libvdsm

A stable and supportable node level API for oVirt



Adam Litke <agl@us.ibm.com>
IBM Corporation

Today's “API”

- Designed to be internal with a single user (engine)
- Complex with lots of special cases and rules
- Inconsistent
- Undocumented
- Designed to work around HTTP/xmlrpc issues
  - 32 bit limit for integers
- Accidental misuse of APIs can cause corruption

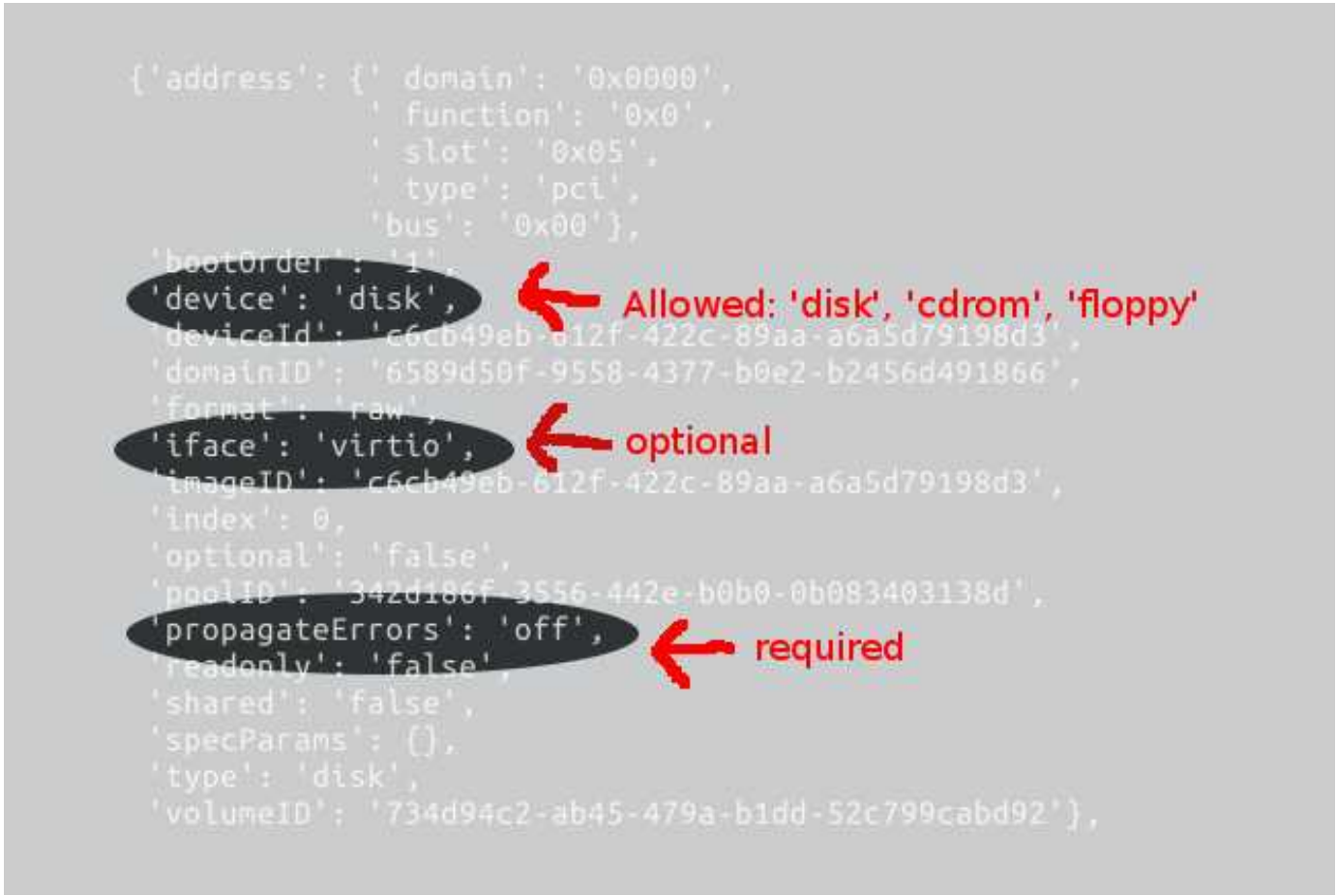
Complexity


```
{'custom': {'device_75cf1037-2e53-4305-aaf0-7b88fa537c66': 'VmDevice {vmId=ebe00e6e-386f-4705-a92f-d5f497bd2a1e,
deviceId=75cf1037-2e53-4305-aaf0-7b88fa537c66, device=ide, type=controller, bootOrder=0, specParams={}, address=
{bus=0x00, domain=0x0000, type=pci, slot=0x01, function=0x1}, managed=false, plugged=true, readOnly=false, alias=ide0}',
'device_75cf1037-2e53-4305-aaf0-7b88fa537c66device_22454b25-
d68d-4b88-860a-2bf57e65935ddevice_5079ed09-65ae-4e4c-886a-3148eff09e09device_ff87943e-b22b-4a40-83ec-6589709259e4':
'VmDevice {vmId=ebe00e6e-386f-4705-a92f-d5f497bd2a1e, deviceId=ff87943e-b22b-4a40-83ec-6589709259e4, device=unix,
type=channel, bootOrder=0, specParams={}, address={port=2, bus=0, controller=0, type=virtio-serial}, managed=false,
plugged=true, readOnly=false, alias=channel1}', 'device_75cf1037-2e53-4305-aaf0-7b88fa537c66device_22454b25-
d68d-4b88-860a-2bf57e65935ddevice_5079ed09-65ae-4e4c-886a-3148eff09e09': 'VmDevice {vmId=ebe00e6e-386f-4705-a92f-
d5f497bd2a1e, deviceId=5079ed09-65ae-4e4c-886a-3148eff09e09, device=unix, type=channel, bootOrder=0, specParams={},
address={port=1, bus=0, controller=0, type=virtio-serial}, managed=false, plugged=true, readOnly=false, alias=channel0}',
'device_75cf1037-2e53-4305-aaf0-7b88fa537c66device_22454b25-d68d-4b88-860a-2bf57e65935d': 'VmDevice
{vmId=ebe00e6e-386f-4705-a92f-d5f497bd2a1e, deviceId=22454b25-d68d-4b88-860a-2bf57e65935d, device=virtio-serial,
type=controller, bootOrder=0, specParams={}, address={bus=0x00, domain=0x0000, type=pci, slot=0x04, function=0x0},
managed=false, plugged=true, readOnly=false, alias=virtio-serial0}'}, 'keyboardLayout': 'en-us', 'kvmEnable': 'true',
'pitReinjection': 'false', 'acpiEnable': 'true', 'emulatedMachine': 'pc-0.14', 'tabletEnable': 'true', 'vmId':
'ebe00e6e-386f-4705-a92f-d5f497bd2a1e', 'devices': [{'device': 'qxl', 'specParams': {'vram': '65536'}, 'type': 'video',
'deviceId': '53bf8fdb-c4a2-4a02-8190-dd5e59836d12', 'address': {'bus': '0x00', ' slot': '0x02', ' domain': '0x0000', '
type': 'pci', ' function': '0x0'}}, {'index': '2', 'iface': 'ide', 'address': {' controller': '0', ' target': '0',
'unit': '0', ' bus': '1', ' type': 'drive'}, 'specParams': {'path': 'Fedora-17-x86_64-netinst.iso'}, 'readonly': 'true',
'deviceId': 'aa56e486-dd13-4fe7-a37c-ee627b1bbb0c', 'path': '/rhev/data-center/342d186f-3556-442e-b0b0-0b083403138d/
ee8e704f-dc3f-418f-806c-8826cb0d07ba/images/11111111-1111-1111-1111-111111111111/Fedora-17-x86_64-netinst.iso', 'device':
'cdrom', 'shared': 'false', 'type': 'disk'}, {'index': 0, 'iface': 'virtio', 'format': 'raw', 'bootOrder': '1', 'poolID':
'342d186f-3556-442e-b0b0-0b083403138d', 'volumeID': '734d94c2-ab45-479a-b1dd-52c799cabd92', 'imageID':
'c6cb49eb-612f-422c-89aa-a6a5d79198d3', 'specParams': {}, 'readonly': 'false', 'domainID': '6589d50f-9558-4377-b0e2-
b2456d491866', 'optional': 'false', 'deviceId': 'c6cb49eb-612f-422c-89aa-a6a5d79198d3', 'address': {'bus': '0x00', '
slot': '0x05', ' domain': '0x0000', ' type': 'pci', ' function': '0x0'}, 'device': 'disk', 'shared': 'false',
'propagateErrors': 'off', 'type': 'disk'}, {'nicModel': 'pv', 'macAddr': '00:1a:4a:16:01:53', 'network': 'ovirtmgmt',
'specParams': {}, 'deviceId': 'f70d6ad8-1ec1-4f82-b5ec-fbcc6356c700', 'address': {'bus': '0x00', ' slot': '0x03', '
domain': '0x0000', ' type': 'pci', ' function': '0x0'}, 'device': 'bridge', 'type': 'interface'}, {'device':
'memballoon', 'specParams': {'model': 'virtio'}, 'type': 'balloon', 'deviceId': 'bf9d331c-924c-443b-8d2c-8955cfc1ac3f'}],
'smp': '1', 'vmType': 'kvm', 'memSize': 1024, 'timeOffset': '0', 'cpuType': 'Opteron_G1', 'smpCoresPerSocket': '1',
'vmName': 'cloudy-1', 'display': 'vnc', 'transparentHugePages': 'true', 'nice': '0'}
```

Inconsistency

```
                'nicModel': 'pv',
                'specParams': {},
                'type': 'interface'},
               {'device': 'memballoon',
                'deviceId': 'bf9d331c-924c-443b-8d2c-8955cfc1ac3f',
                'specParams': {'model': 'virtio'},
                'type': 'balloon'}],
 'display': 'vnc',
 'emulatedMachine': 'pc-0.14',
 'keyboardLayout': 'en-us',
 'kvmEnable': 'true',
 'memSize': 1024,
 'nice': '0',
 'pitReinjection': 'false',
 'smp': '1',
 'smpCoresPerSocket': '1',
 'tabletEnable': 'true',
 'timeOffset': '0',
 'transparentHugePages': 'true',
 'vmId': 'ebe00e6e-386f-4705-a92f-d5f497bd2a1e',
 'vmName': 'cloudy-1',
 'vmType': 'kvm'}
```

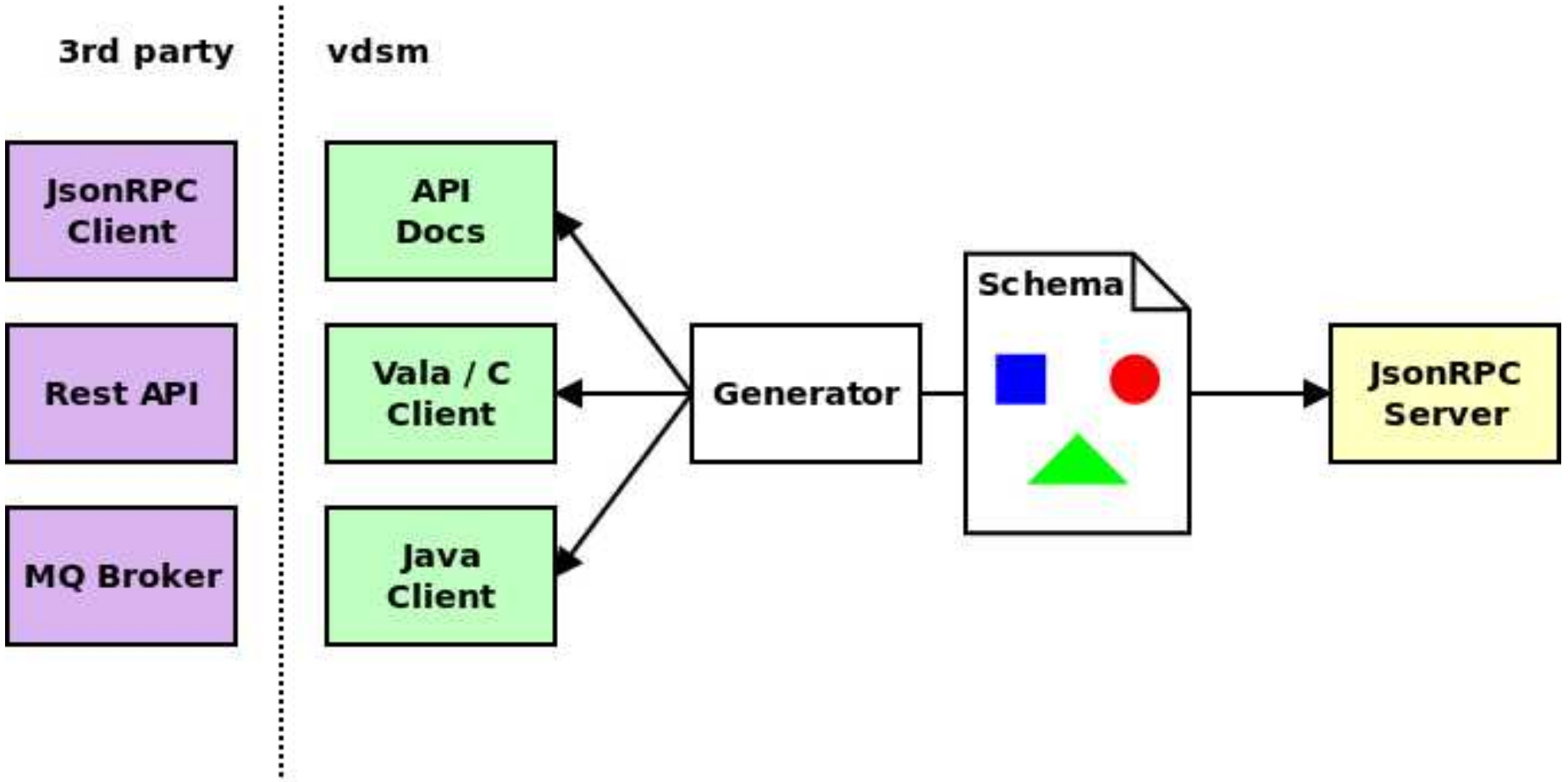
Ambiguity



The Goal

- Create a better API that is
  - Stable but evolving
  - Easy to consume from many diverse clients
  - Always well documented
  - The only supported vdsm interface
  - A foundation for 3rd party extensions such as
    - Rest API
    - MQ broker

Libvdsm components



Schema

- Authoritative definition of API commands and types
- Basic types
  - int, uint, bool, str, double, enum
- Container types: list and map
- Object types
- Commands
  - Class, name, parameters, return value
- Errors
- Documentation
  - Each object and command is documented

Schema

```
##
# @VmTicketConflictAction:
#
# An enumeration of consequences if another user is connected to a display.
#
# @disconnect:  Disconnect the current user
#
# @keep:        Change the password but keep the current user connected
#
# @fail:        Fail the operation and do not change the password
#
# Since: 4.10.0
##
{'enum': 'VmTicketConflictAction', 'data': ['disconnect', 'keep', 'fail']}

##
# @VM.setTicket:
#
# Set the password to be used to connect to a VM display.
#
# @password:            The desired connection password
#
# @ttl:                 The number of seconds before the password expires
#
# @existingConnAction:  Indicate what to do with any existing connections
#
# @params:              Arbitrary key:val pairs that will be passed to hooks
#
# Since: 4.10.0
##
{'command': {'class': 'VM', 'name': 'setTicket'},
 'data': {'password': 'str', 'ttl': 'int',
          'existingConnAction': 'VmTicketConflictAction',
          'params': 'StringMap'}}
```

Server

- Based on JSON-RPC specification
  - The raw protocol will be a supported interface
- Dynamic dispatching to vdsm by using the schema
  - Currently we require code to handle schema exceptions
  - In future, server will depend upon the schema only
- Planned improvements
  - Events
  - SSL

The protocol

- All messages

```
<size 64-bit big-endian><json message body>
```

- Request

```
32{'id': 1, 'method': 'Host.ping'}
```

- Response

```
25{'id': 1, 'result': null}
```

- Event

```
50{'method': 'VmCreated', 'params': {'vmId': 'xxx'}}
```

Schema Exceptions

- The schema does not match the internal API
  - vdsm functions must maintain compatibility with xmlrpc
- Server contains code to translate between the two
- Parameter formats
  - Numbers and booleans represented as strings
- Parameter and return value mangling
  - VmDevice: type → deviceType
- New / virtual APIs
  - VM.getInfo()

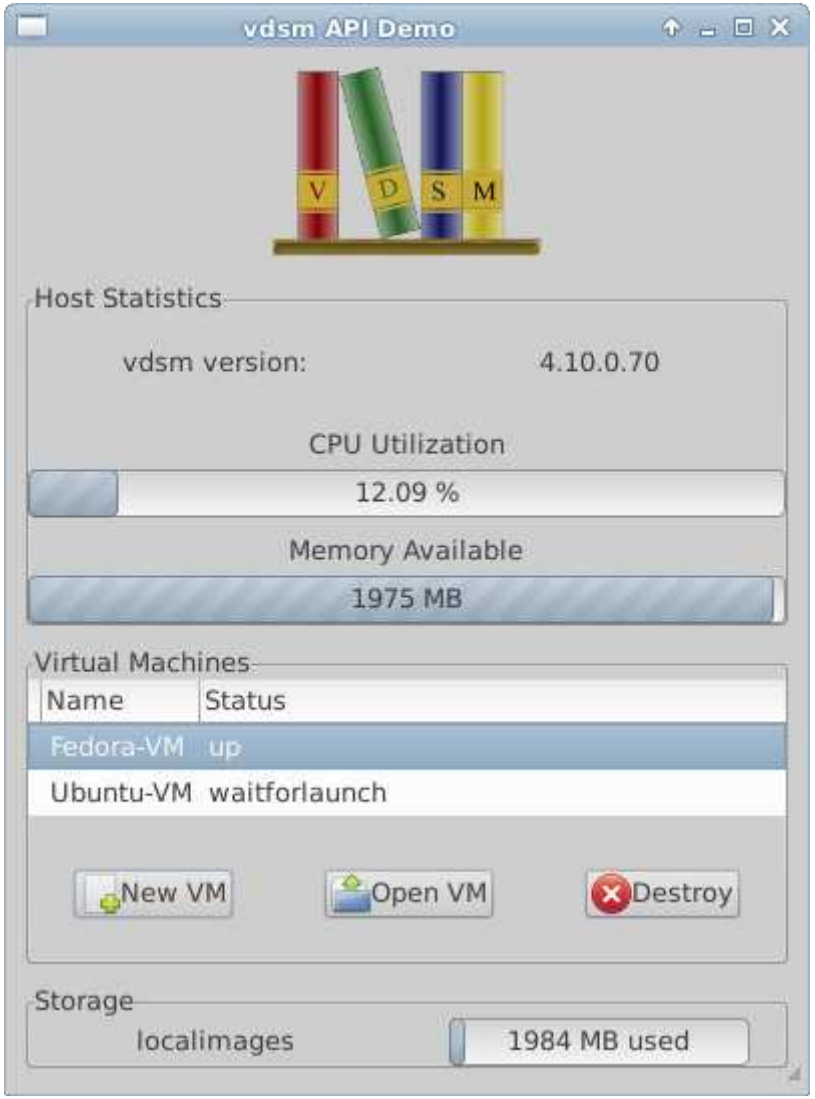
Code / documentation generation

- Single touch – one place to edit when adding APIs
- API documentation generated entirely from schema
  - Comment block provides human-readable information
  - Symbol definitions provide types and cross-references
- Schema complete enough to generate client libraries
  - Classes and commands
  - Primitive and structured types
  - Parameters (optional vs. required)
  - Could be extended to note defaults for primitive types

Client C library

- Written in Vala
  - Language is designed to simplify the use of gobjects
  - Small static file contains protocol implementation
  - The rest is generated at build time by a python script
- Provides Vala and C support natively
- Python bindings available with gobject-introspection
- Java interface is on the TODO list
  - Not complex: client doesn't contain much logic

Demo



http://wiki.ovirt.org/w/images/a/ac/Libvdsm-demo.tgz

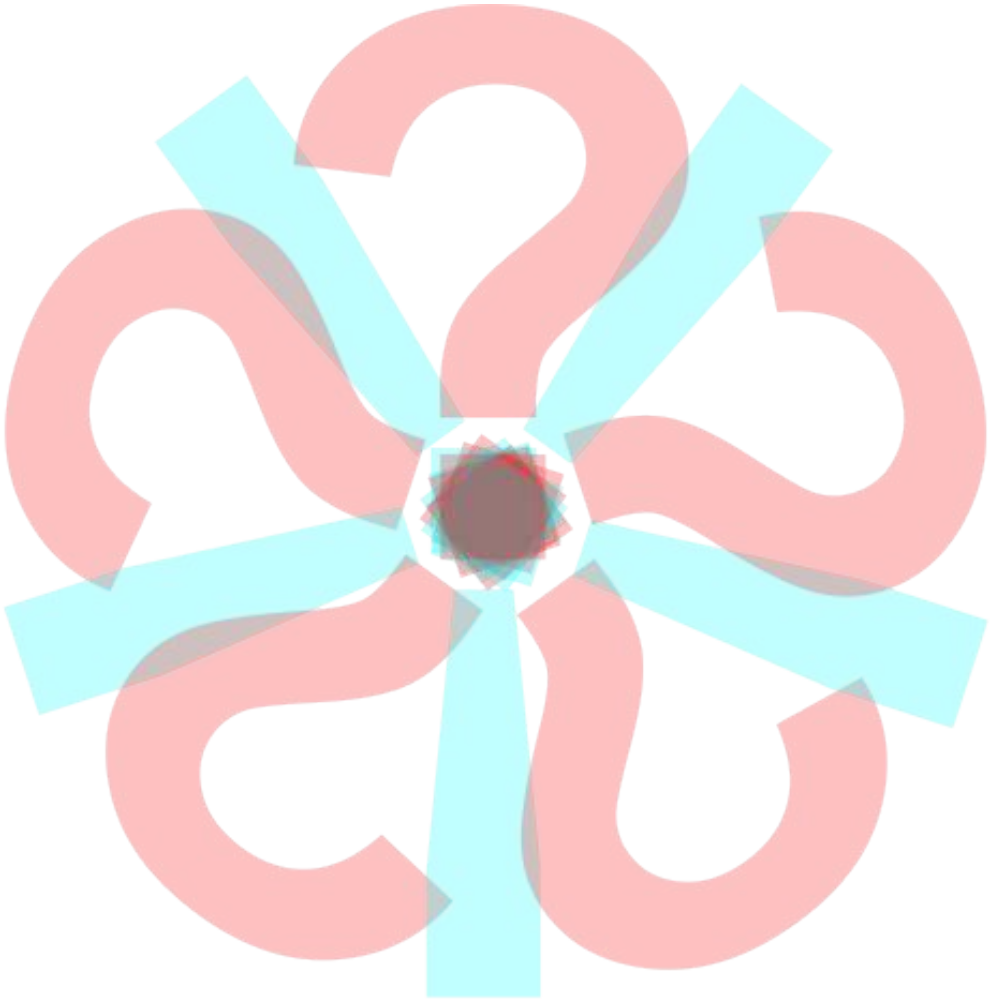
Rules for evolving the API

- We'll apply these rules once the API becomes stable
- Follow the rules for any C API
- Do not change or remove function signatures
  - Do use a flags parameter to extend function behavior
  - Do add new functions when parameters must change
  - Functions may be removed with a major version bump
- Do not remove fields from complex types
  - Do add new fields if you bump the minor version
- Do use capabilities to indicate semantic changes

oVirt integration plan

- libvdsm 0.1.0:
  - Based on current vdsm API
  - Generate Java bindings and migrate ovirt-engine
  - Remove xmlrpc interface
  - No commitment for backwards compatibility
- libvdsm 1.1.0:
  - Clean up API and remove schema exceptions
  - Optional: One-time compatibility break for libvdsm-1.0
    - Deprecate functions
    - Change complex types

Q & A



Links

- http://www.jsonrpc.org/specification
- https://live.gnome.org/Vala
- https://live.gnome.org/GObjectIntrospection
- Current patches
  - project:vdsm, topic:libvdsm
- Demo program
  - http://wiki.ovirt.org/w/images/a/ac/Libvdsm-demo.tgz