Community-driven onboarding

Mark Shuttleworth & Arno van Huyssteen  (Canonical)
Bootstrapping a community to create and maintain OSM VNFDs for both open source and proprietary network functions
System Integrator Engagements
SI Engagement Goals

- Canonical will appoint 3 SIs for open source VNFD creation
- Target high-visibility workloads, both open and proprietary
- Bootstrap a portfolio of OSM-compatible VNFDs
- Establish best practices for VNF onboarding in OSM
- Address issues in OSM uncovered by testing
- Accelerate OSM evaluation and adoption
Community Engagement Method
Charm & VNFD Bootstrapping
3 paid SI engagements

Slack Channel
#vnf-onboarding

Charm Training
Online & On Site
VNF Onboarding Track

1 call per week
Europe Time

Charm Development Track

3 calls per week
Americas, Europe, Asia

Canonical Centers of Excellence

Slovakia, Brazil, Taiwan
Reusable Code for Operations
Operating workloads

Operator instance

VNFD

Workload

OSM
VNFD

Metadata

Operations package - "Charm"
- Lifecycle
- Configuration
- Operation
- Integration

DECLARATIVE

CODE
Unified operator framework for K8s and legacy workloads
Python Operator Framework
“Make it easy to create and maintain a Kubernetes operator in pure Python”
Why Pure Python?

- High-level
- Widely known
- Low entry barrier
- Good for writing integration code
- Simple debugging & testing
- Cross-architecture
Python Operator Framework

• Class hierarchy
  • Charm; Event; Endpoint; StoredState; Action; Model; Unit ...
• Event system;
• Persistence layer;
• Minimal dependencies.
Upstreaming Charmed OSM
OSM Operations in OSM
Shared Engine for OSM Distributions

- Global market for MANO is too large for one company
- Enable many SI and NEP OSM distributions
- Efficient mechanisms for customisation and differentiation
- Collaborate on operational mechanisms
- Reusable integrations for third-party solutions
- Optional escalation support from Canonical team
Find the Canonical OSM team at:

arno.van.huyssteen@canonical.com