

RED HAT  
**SUMMIT**

**LEARN. NETWORK.  
EXPERIENCE OPEN SOURCE.**

June 11-14, 2013  
Boston, MA

RED HAT  
**SUMMIT**

# Getting Ready for systemd, the new Red Hat Enterprise Linux 7 Service Manager

Lennart Poettering, Sr. Software Engineer, Red Hat  
Kay Sievers, Sr. Software Engineer, Red Hat  
June 14<sup>th</sup>, 2013

systemd is a *service* manager

systemd is a *system* manager

systemd is more than just an init  
system

# 11 Kinds of „Units“

Services, Sockets, Targets,  
Devices, Mounts, Automounts,  
Snapshots, Timers, Swaps, Paths,  
Slices

# Dependency System Between Units

## Example:

1. Wait for Block Device
2. File System Check for Device
3. Mount File System
4. Do This for All Necessary File Systems

- Minimal Boot Times
  - Reliability
  - Debuggability

# Features:

Keeping Track of Processes

„ps“ With Service Information

Killing of Services

→ cgtop, cgls

Features:  
Autospawn

Resource Management  
Slices

Features:

*No Log Messages Ever Lost*

All stdout+stderr Goes to Syslog

Early Boot Logging, Kernel Logging

# Features:

## Hierarchical Watchdog

# Features:

Socket Activation

Hardware Activation

Bus Activation

Timer Activation

Path Activation

Features:

- Debugging Tools
- Profiling Tools
- Introspection Tools
- Predictable Results

# Features: Extensibility

State Exposed on the Bus  
Write Scripts against the Bus  
Interfaces

# Compatibility:

## 99% Compatibility with SysV Init Scripts

# Writing Service Files

Declarative Description  
Not a Turing-complete  
Programming Language

# Writing Service Files

```
[Unit]
```

```
Description=FooBar
```

```
[Service]
```

```
ExecStart=/usr/bin/foobard -D
```

```
Type=forking
```

```
[Install]
```

```
WantedBy=multi-user.target
```

# Writing Service Files

But – Service Files are shipped upstream in most cases

# Execution Parameters

Nice Level, Resource Limits,  
Environment Variables, IO Priority,  
CPU Priority, CPU Affinity, Timer Slack, chroot(),  
chdir(), setuid(), setgid(), OOM Score, umask, ...

# Security

Capabilities, Private Network,  
Device Access, Read-  
Only/Inaccessible Directories,  
Private /tmp, System Call  
Filters, ...

# Control Groups

Per-Service Resource Settings

CPU, Memory, IO, ...

Persistent and at Runtime

# Benefits over sysvinit:

Sysvinit loses log messages, can't respawn daemons, doesn't record runtime data, loses all daemon context during runtime, can't even kill services cleanly, ...

# Core Component for RHEL 7

Or try it out today  
on Fedora 15 and newer!

That's All!

Meet us in the lobby!

Thank you for listening!