Elettra Sincrotrone Trieste

Elettra
Sincrotrone
Trieste

CERTIFIED MANAGEMENT SYSTEMS
CERTIQUALITY
UNI EN ISO 9001:2015
UNI CEI EN ISO/IEC 27001:2022
UNI EN ISO 45001:2023
UNI CEI EN ISO 50001:2018

Voltumna Linux
A custom, layered, immutable distribution for Tango-based accelerator control systems

Q&A at slido.com #3609271

The challenge: an archaeological fleet

- ✓ Same beamline, twenty-five years of hardware:
  - VME PowerPC SBCs (MVME5100/7100/2500) from 1999
  - x86 rack servers, NUCs, UPBoards, Aaeon, Supermicro, Dell
  - ARM controllers (BeagleBone, BeagleBone AI)
  - FPGA SoCs (Cyclone V, Arria 10, in-house Dinet)
- ✓ Each generation arrived with its own kernel, libc, init, quirks
- ✓ Multiplied by 25 years of operations:
  - operations model that does not scale
  - development environment that does not scale
  - long-term support story that is, essentially, "good luck"

The idea: one recipe, many Linuxes

- ✓ Voltumna: a Yocto / OpenEmbedded meta-distribution
  - fully owned and maintained by the Controls group
  - second generation after the Flop prototype (ICALEPCS 2015)
- ✓ Not one Linux — a layered recipe for producing many Linuxes
  - same source tree, same toolchain, same Tango libraries
  - identical from a 1999 MVME5100 to a 2024 Xeon server
- ✓ Production today on the Elettra storage ring and on the FERMI free-electron laser

Layered architecture

- ✓ Distribution layers
  - meta-voltumna (core)
  - meta-elettra
  - meta-ess
  - meta-tango
- ✓ BSP / vendor layers
  - meta-artesyn (legacy VME/PPC)
  - meta-intel, meta-arm, meta-ti
  - meta-intel-fpga
- ✓ Tooling layers
  - meta-clang
  - meta-dpdk
  - OpenEmbedded core
- ✓ Every layer is a Git submodule of meta-voltumna
- ✓ Every release image is identified by a Git tag
- ✓ Reviews, bisection, blame, tagging and CI apply naturally

Distributions and flavours

- ✓ Purpose-built distributions
  - ec — Equipment Controller
  - ccd — CCD readout
  - ebpm — Elettra Beam Position Monitor
  - imrf — RF Interlock Module
  - a2720 — custom power-supply controller
- ✓ Three flavours per distribution
  - SRE — Software Runtime Environment
    - immutable, no compilers, no debuggers
    - minimal attack surface, production hosts
  - SDE — Software Development Environment
    - full native target development
  - SDK — cross-compiling, runs on Ubuntu hosts
    - short build cycles on workstations

What we get technically

- ✓ Reproducibility down to the byte
  - every recipe pins a source URL and SHA-256
  - every layer is a Git submodule at a fixed commit
  - BIOS, NIC and adapter firmware captured in the image
  - rebuild the exact image months or years later
- ✓ Immutable images, atomic upgrade and rollback
  - /usr is read-only; upgrades replace the image
  - rolling back a misbehaving node is a reboot
- ✓ Cross-architecture homogeneity
  - PowerPC, x86_64, ARM, FPGA SoC
  - one source tree, one Tango model, many CPUs
- ✓ Fine-grained control of the low-level stack
  - custom MACHINE files, custom kernel configs
  - isolcpus, cgroups, vfio, DPDK for partitioning
  - Use specific ISA

Real-time, and a first-class Tango stack

- ✓ The same image runs general-purpose and hard-real-time loads
  - isolcpus + cgroups reserve CPUs from the scheduler
  - VFIO + DPDK hand PCIe devices to user-space loops
  - used by the Elettra 2.0 real-time platform (Gaio et al., ICALEPCS 2023)
- ✓ meta-tango makes Tango a first-class citizen
  - cppTango, PyTango, jive, astor packaged as recipes
  - device-server skeletons ship with the image
  - a Tango device server is just another recipe
- ✓ Built once, runs identically on every supported board

Deployment: diskless, atomic, reversible

- ✓ Front-ends are diskless and network-booted
  - small set of virtualised service hosts: DHCP / TFTP / NFS / Tango DB
  - front-end pulls a read-only image at boot
- ✓ Upgrading a controller = replacing one image with another
  - full and incremental .upd files served from the deploy server
  - rollback to the previous tag is a reboot
- ✓ Replacement of a failed controller is a hardware operation
  - plug in, power up, identical state
- ✓ Companion tool INAU (PCaPAC 2022) extends the same Git-based flow to Tango device servers

Operational and strategic benefits

- ✓ Collapsed maintenance surface
  - same init, network stack, libc, SSH policy everywhere
- ✓ Security as a side effect of minimalism
  - no compiler, no debugger, no package manager on target
  - root login disabled, filesystem read-only
  - a compromised node loses its delta on next reboot
- ✓ No vendor lock-in, no subscription cliff
  - dependencies: upstream Yocto + Git. That is all.
  - know-how stays in the controls group
- ✓ Institutional knowledge captured in Git
  - every tweak is a recipe or a patch
  - survives staff turnover

Elettra Sincrotrone Trieste

Why immutable? The mutable-distro failure mode

- ✓ A classic apt / dnf system is mutable global state
  - its exact contents depend on when it was installed and last updated
- ✓ Update two identical boxes a month apart → different package sets
  - configuration drift: each becomes a one-off “snowflake server”
- ✓ “What exactly ran on front-end 17 during March’s beam time?”
  - on a mutable fleet, the honest answer is “we’d have to reconstruct it”
- ✓ Voltumna: state = one Git tag, /usr read-only, atomic rollback
  - drift is not managed — it is structurally impossible

Elettra Sincrotrone Trieste

Voltumna vs Debian vs Ubuntu / snap

Axis	Voltumna	Debian stable	Ubuntu + snapd
Unit of state	Tagged image (incl. firmware)	Per-package set	Packages + auto-refresh snaps
Who controls when	Controls group, explicit	Admin runs apt	snapd, ~4×/day by default
After release	What we ship, when we ship	Security fixes only	Rolling snap revisions
Reproducibility	Bit-exact whole image	Build-time (≥95%)	+ opaque snap revisions
Rollback	Reboot to previous tag	None at OS level	Prior snap revision
Lifecycle	1999→2024, one tree	~5 yr then EOL	~5 yr + churn

Elettra Sincrotrone Trieste

The honest trade-offs

- ✓ Packaging breadth & QA gap
  - Debian has 65,000+ packages and a security team; Yocto/OE leaves CVE policy to us
- ✓ Support window & CVE burden
  - Yocto ~1 yr, LTS “Scarthgap” ~4 yr, commercial ~10 yr — we are our own security team
- ✓ The “frozen after release” double edge
  - big distros ship only security fixes — we match that by choice, not by inheritance
- ✓ Cost of ownership
  - build-infra, BitBake curve, fewer external eyes, bus-factor — eased by everything-in-Git

Elettra Sincrotrone Trieste

Why the trade-off favours a controls group

- ✓ Mainstream distros optimise for the average desktop / server
  - our fleet is the opposite: 25-yr HW span, hard real-time, diskless, requalification
- ✓ What immutability “costs” — you own everything — is exactly what buys
  - reproducibility down to firmware, atomic one-reboot rollback
  - cross-facility reuse without forking (meta-elettra ↔ meta-ess)
- ✓ Debian gives stability by freezing; Ubuntu convenience by auto-updating
  - Voltumna gives determinism by construction — state is a name, not a history
- ✓ Verdict: we trade breadth and free QA for total, reproducible control
  - over a fleet a big distribution was never designed to serve

Cross-facility reuse, not forks

- ✓ The A2720 power-supply controller runs at both Elettra and the European Spallation Source
- ✓ Same hardware, same recipes, byte-identical lower layers
  - only meta-elettra is swapped for meta-ess
  - no patch sets to maintain, no divergent toolchains
- ✓ The distribution becomes a collaboration vehicle, not a fork point
- ✓ Anyone in this room can pick up meta-tango and build on it

Elettra Sincrotrone Trieste

Take-aways

- ✓ In production at Elettra storage ring and FERMI free-electron laser
- ✓ Supports hardware from 1999 (MVME5100) to 2024 (FPGA SoC) from a single source tree
- ✓ Bit-exact reproducibility, including BIOS and adapter firmware
- ✓ Atomic upgrade and one-reboot rollback on diskless front-ends
- ✓ Hard real-time and Tango on the same image
- ✓ Open source: github.com/voltumna-linux
  - meta-tango is reusable today
- ✓ If you have a fleet of front-ends and a chronic upgrade headache — let us talk afterwards

Elettra
Sincrotrone
Trieste

CERTIFIED
MANAGEMENT SYSTEMS
CQY
CERTIQUALITY
UNI EN ISO **9001:2015**
UNI CEI EN ISO/IEC **27001:2022**
UNI EN ISO **45001:2023**
UNI CEI EN ISO **50001:2018**

Thank you!