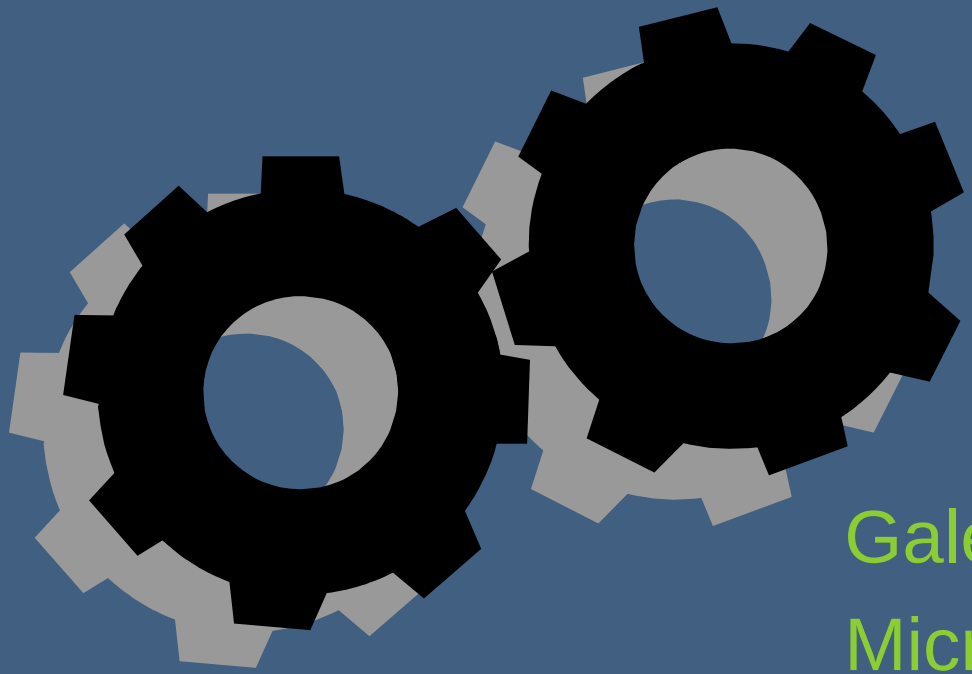
Exploring new OS Research through Singularity



"... it is impossible to predict how a singularity will affect objects in its causal future."

- NCSA Cyberia Glossary

Galen Hunt

Microsoft Research

Singularity

- New Research OS
  - Platform for OS research agenda
  - Common laboratory for cross-group research into software development w/ PPRC. (NP)
- Managed code all the way down.
  - Written in SpeC#
    - Superset of C# (formerly A#)
    - Pre and Post Conditions, Invariants, etc.
  - Bartok runtime (for MSIL) on bare metal.

Abstract Instruction Set

- Safe MSIL is the universal OS binary interface
  - Language Safety and Isolation
  - ISA neutral and Structured Metadata
- OS controls translation to native code.
  - Generate qualitatively different code from source MSIL
    - Translate for security vs. performance:
      - buffer overflows, stack walking, NX, SFI, etc.
    - Ubiquitous on demand instrumentation:
      - asserts, profiling, debugging, Vulcan, etc.
  - Rethink traditional hardware security boundaries.
    - Inline privileged operations into non-kernel code.

Ubiquitous Metadata

- Make metadata a first class, protected OS feature.
  - Hang extensible metadata off all named OS entities.
    - processes, threads, handles, files, security principals, etc.
  - Strong typing and schemas for OS interfaces
    - registry, ioctls, /proc, etc.
  - Unify OS metadata APIs
    - for enumeration, change notification, security, etc.
- Add OS abstractions to bridge metadata gaps.
  - (Program → Process) is like (Class → Instance)
- Enable complete static analysis of OS *and* applications
  - at compile and deployment time
  - for verification, optimization, versioning, etc.

Rebuild The OS Security Model

- Remove drivers from the TCB
  - Reify driver/service trust relationships.
  - Hardware to ensure memory safety (DMA “MMU”).
- Remove Administrator from the TCB
  - Reify user trust relationships.
  - Orthogonal Management
    - Kernel (OS Vendor) as Trusted Third Party
  - What is “complete” solution for NGSCB scenarios?
- Simplify security management
  - Deep isolation mechanisms.
  - Eliminate partial redundancies between OS and Runtime.

Reengineer the OS Development Process

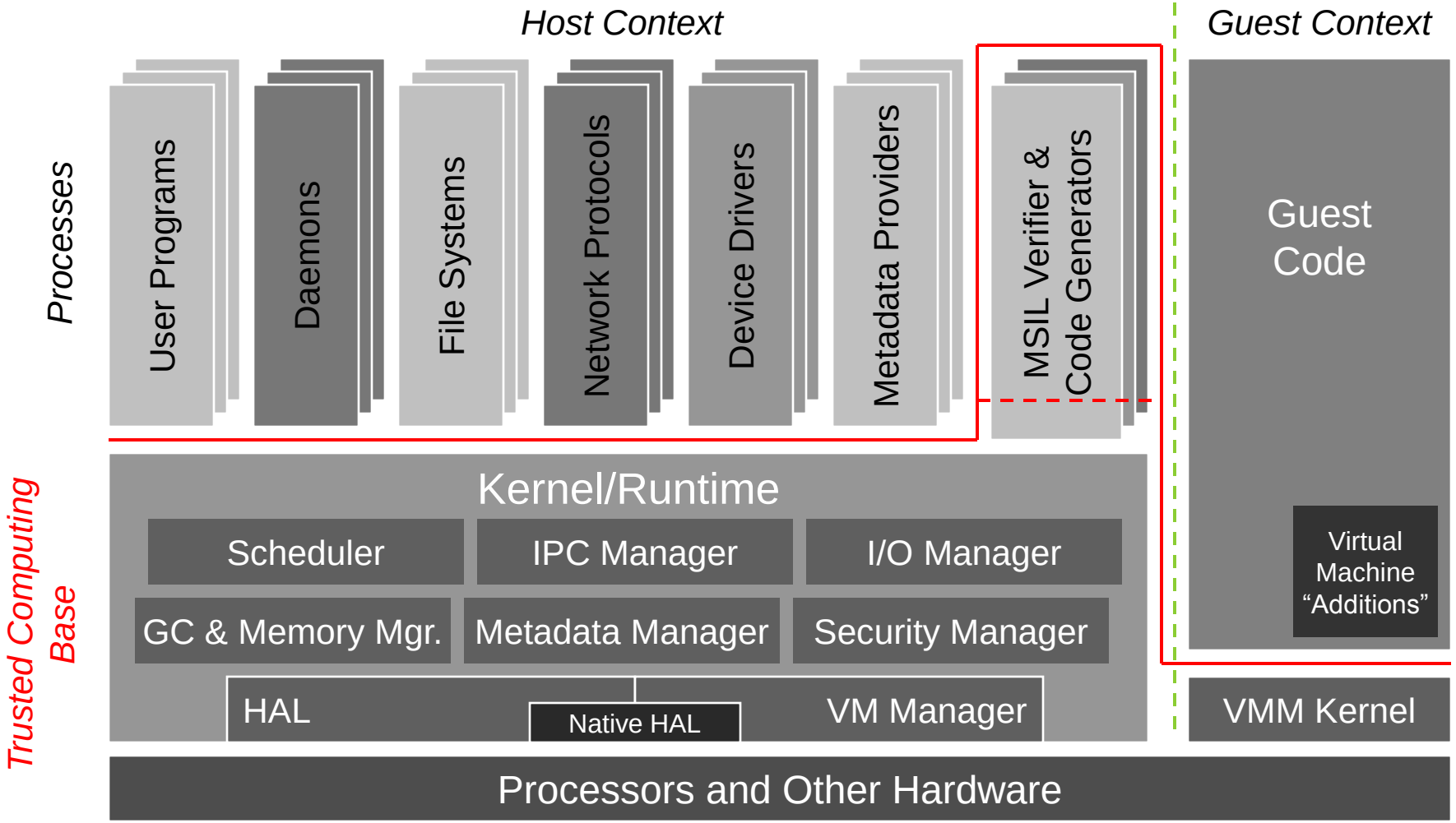
- Software 2010 Grand Challenge
  - "Routine to build secure, reliable, and maintainable applications & systems"

- NP (New Project with Name Pending):
  - Cross group research project
    - OS, SWIG, SBT, FSE, ACT, etc.
  - Explore co-evolution of OS and tools.
    - Structure OS code to meet tools half way.
    - Push tools research into design and deployment phases.
  - Cooperating independent teams
    - focus on own research domain
    - share problems, solutions, people, ideas with other groups

Target Scenario

- eHome digital-convergence
  - Diverse environment with little platform legacy
    - Smart remotes, set top boxes, and media servers.
    - Devices come and go. Storage comes and goes.
    - Wide variety of form factors. One OS.
    - Soft real-time and distributed requirements.
  - Trustworthy, open software platform.
  - Zero on-site human administration.

Architecture



Demo

Truth in Advertising

- What you just saw:
  - Bitmaps (1 per slide) statically linked to Singularity kernel
    - Rendered with safe device drivers.
  - Compiled MSIL running on bare metal.
  - Network boot of Singularity from minidump via PXE.
- What you didn't see ('cause it ain't there yet):
  - Threads or a scheduler
  - Interrupts (drivers are polling)
  - Virtual memory, IPC, or a network stack
  - Metadata
- For more info, see http://singularity.