Singularity

General-purpose Computing Platform (PC)

- Software is difficult to install, maintain, and administer
- Applications interact in complex, unpredictable ways
- Almost no users understand computers or software and so react naively to unexpected behavior
- System administration is costly and unavailable to most
- Contemporary OS designs tend to favor performance over
  - reliability,
  - security,
  - predictability

Singularity Summary

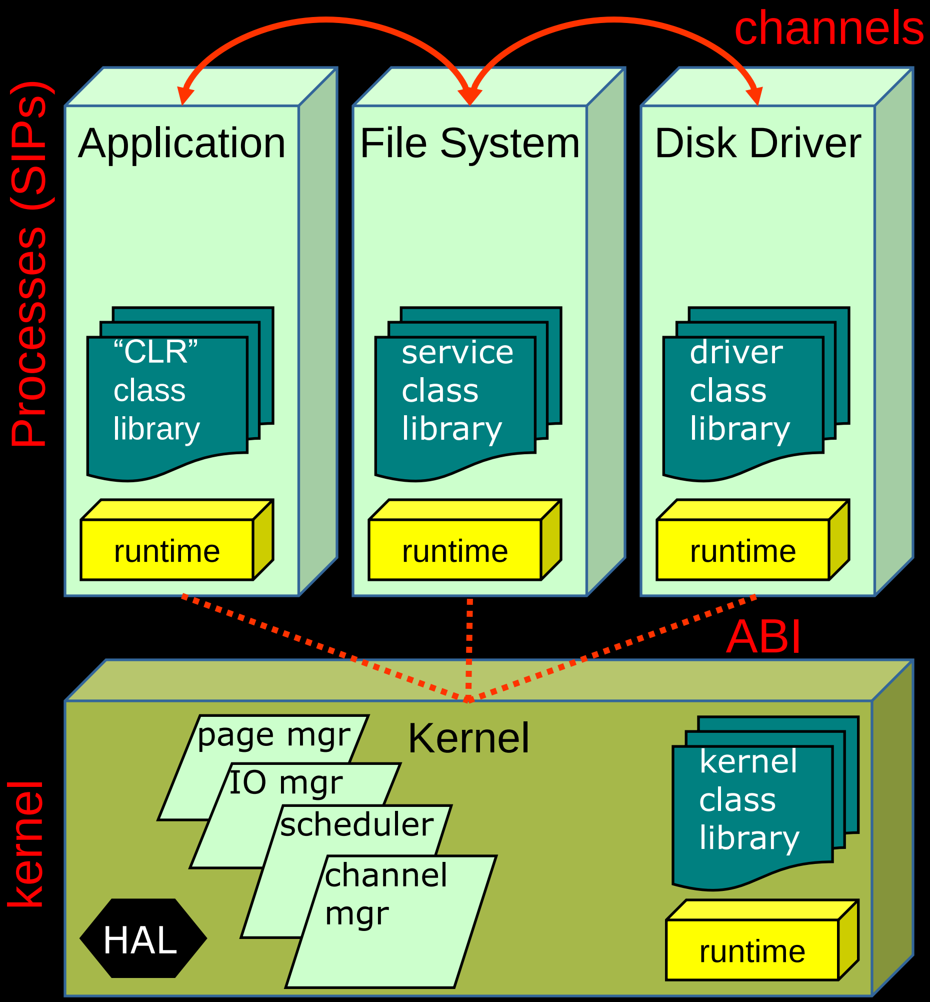
- Advances in languages, compilers, and tools open the possibility of improving software
- Singularity uses these advances as a basis to build more reliable systems and applications
- Systems built on Singularity expand software delivery opportunities

Key Aspects of Singularity

- Software-isolated processes (SIPs)
  - inexpensive isolation — memory, communications, failure — boundaries
  - OS manages and reclaims resources
  - “closed world” for program analysis
  - single isolation and extension model for all parts of application and system
- Merge OS and language runtime (VM/CLR)
  - prohibit unsafe code
  - remove duplicative APIs and security abstractions
  - fast, lightweight managed code run-time system
  - typed assembly language (TAL) reduces trusted computing base
- Language extensions to improve reliability
  - Spec#/Sing# specifications and verification
  - channel contracts
  - explicit and verified resource usage and reclamation
- Not Windows successor!

Detailed Architecture

- Microkernel
  - apps, extensions, services, and drivers are all processes
  - HAL w/ PIC, RTC, timer, and console output
- Closed processes (SIPs)
  - no shared memory
  - no dynamic code loading
  - no dynamic code generation
- IPC via channels w/ contracts
- Abstract instruction set
  - type safe, memory safe MSIL
  - all third-party code is safe
  - layer of indirection
- Well-defined, strongly-versioned application binary interface (ABI)


channels
Processes (SIPs)
Application
“CLR” class library
runtime
File System
service class library
runtime
Disk Driver
driver class library
runtime
ABI
kernel
Kernel
page mgr
IO mgr
scheduler
channel mgr
HAL
kernel class library
runtime