SAND2013-9471P

A Use Case Approach to Deriving Power API Requirements

Sue Kelly and Jim Laros

Sandia National Laboratories

Ryan Elmore, Steve Hammond, and Kris Munch

National Renewable Energy Laboratory

Sandia National Laboratories

Exceptional service in the national interest

HPC Operations Review

November 5, 2013

U.S. DEPARTMENT OF ENERGY

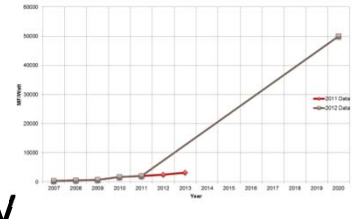
NNSA National Nuclear Security Administration

ASC

NREL NATIONAL RENEWABLE ENERGY LABORATORY

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Why Develop a Power API for HPC?



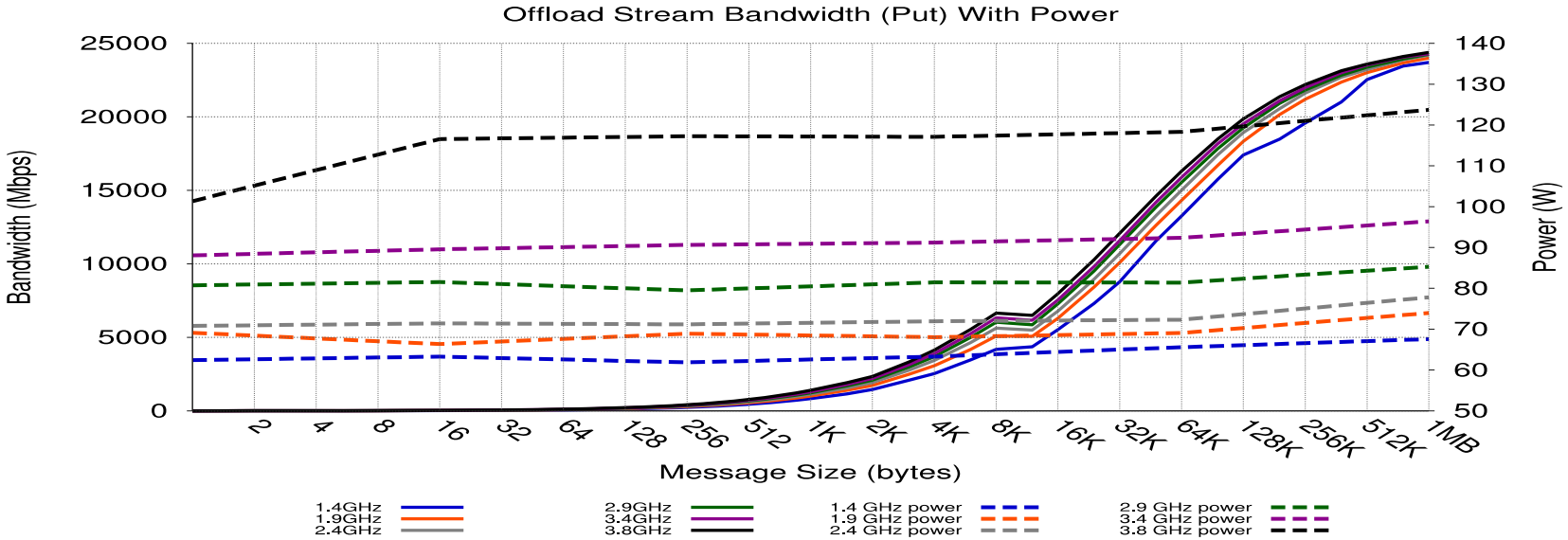
- Anticipated HPC computational needs within reasonable power constraints require significant advances in hardware power efficiency
- To achieve greatest efficiency, software at many levels will need to coordinate and optimize the hardware power features
  - Commodity pressures will drive useful innovations
  - Our efforts are distinguished by HPC requirements at scale
  - We found no existing API specification
- Goal is to create a generic power API for general adoption within the HPC community
- Our intent is to
  - help ASC field machines in the future within reasonable power budgets
  - contribute to the national effort on Exascale, or at least extreme scale scientific computing

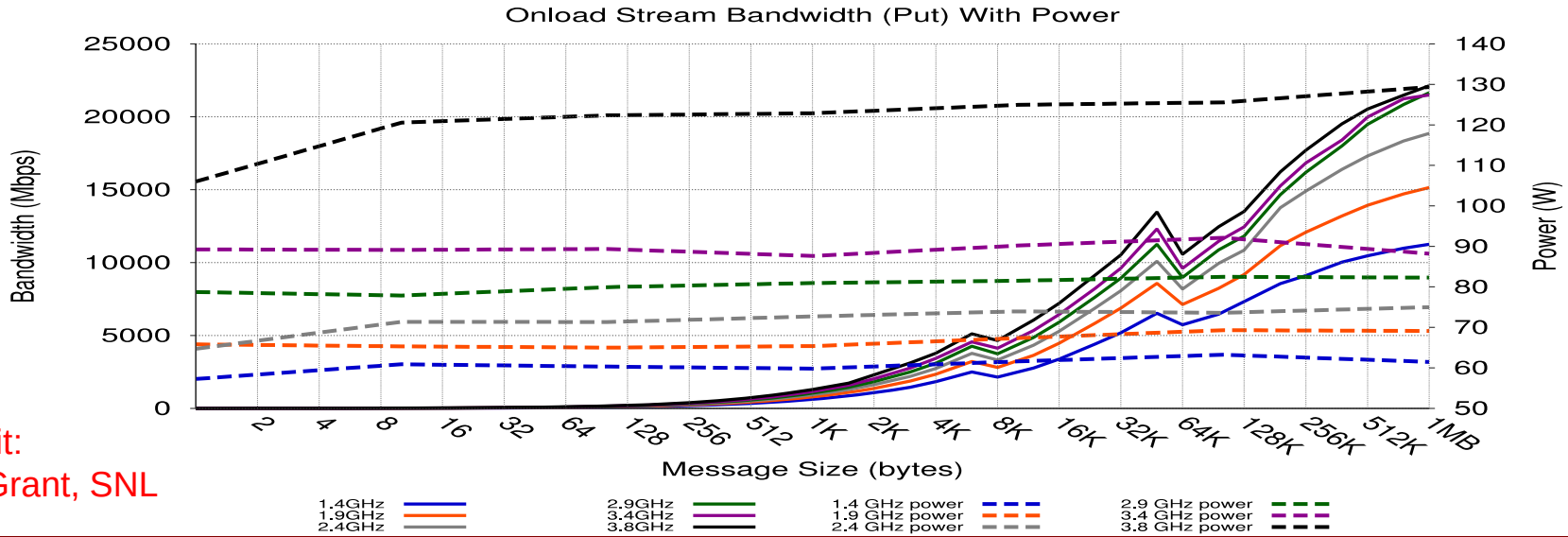
Example Scenarios

- A job is entering a checkpoint phase. Application requests a reduced processor frequency during the long I/O period.
- Developer is trying to understand frequency sensitivity of an algorithm and starts a tool that analyzes performance and power consumption while the job is running.
- Data Center has a maximum of capacity of nn MW. One HPC system is down for extended maintenance. Other systems can have a higher maximum power cap.
- Power company charges more for electricity during the day than it does at night. Schedule jobs by allocating both node and power resources accordingly.

IB power usage - offload vs. onload cards

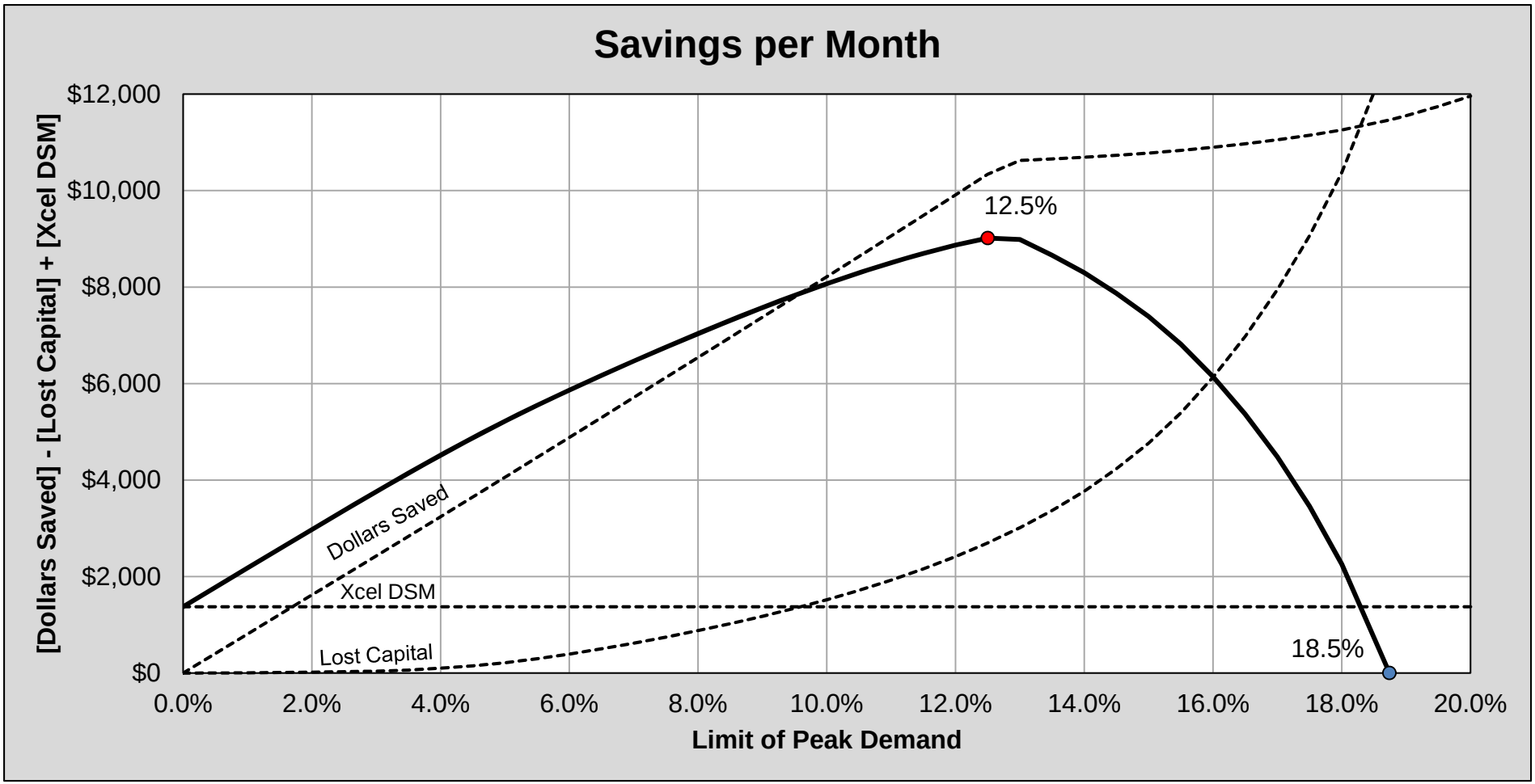
Preliminary Study [2]





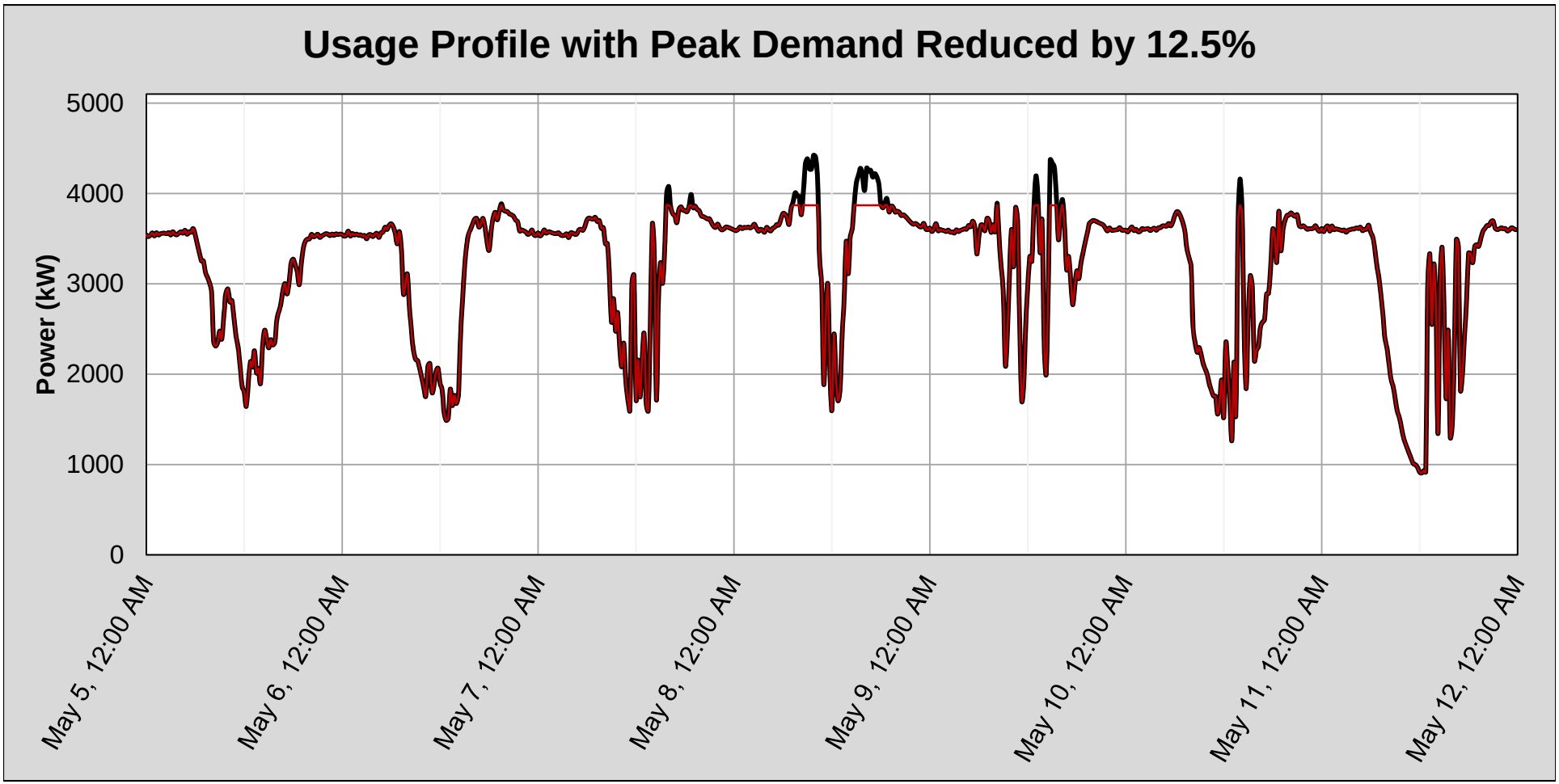
[2] Credit:
Ryan Grant, SNL

Finding Optimal Savings.



Credit: Steve Hammond
National Renewable
Energy Lab

Balance utility savings vs. lost capital.



Credit: Steve Hammond
National Renewable
Energy Lab

Approximate Monthly Savings $9K

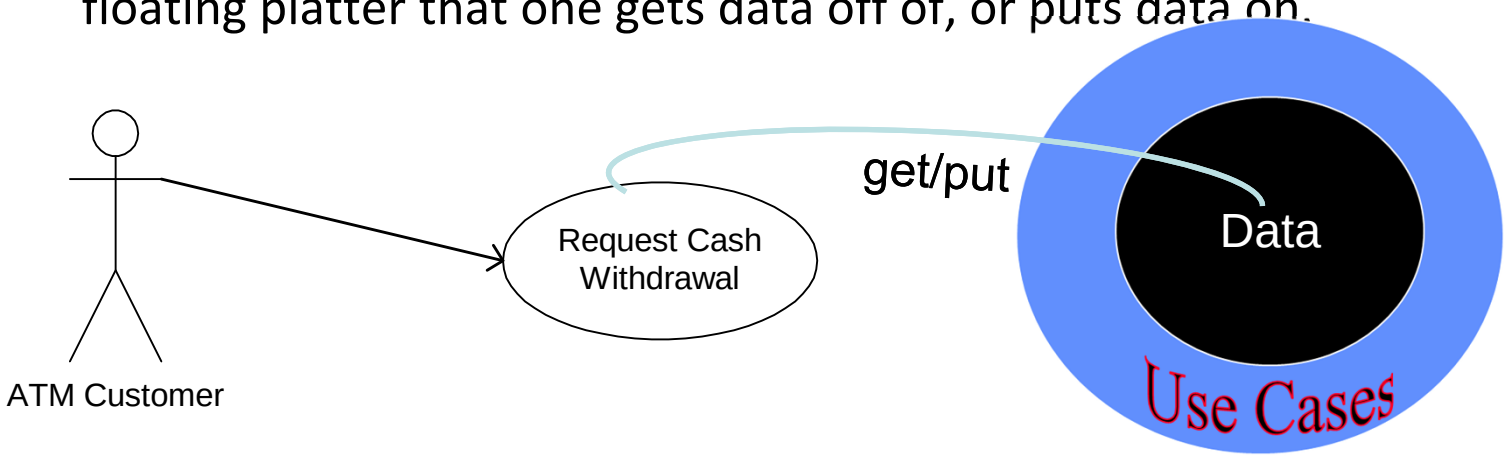
Power API – *A Use Case Approach*

- Prior to specifying an API from scratch, we elected to create formal use cases[1] to model the ways power measurement and control capabilities will be used in HPC systems.
- Use case approach is used to define **SCOPE, INTERFACES,** and **DATA REQUIREMENTS.**
- Generally more intuitive than a laundry list of specifications.

[1]Ivar Jacobson. *Object Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1992.

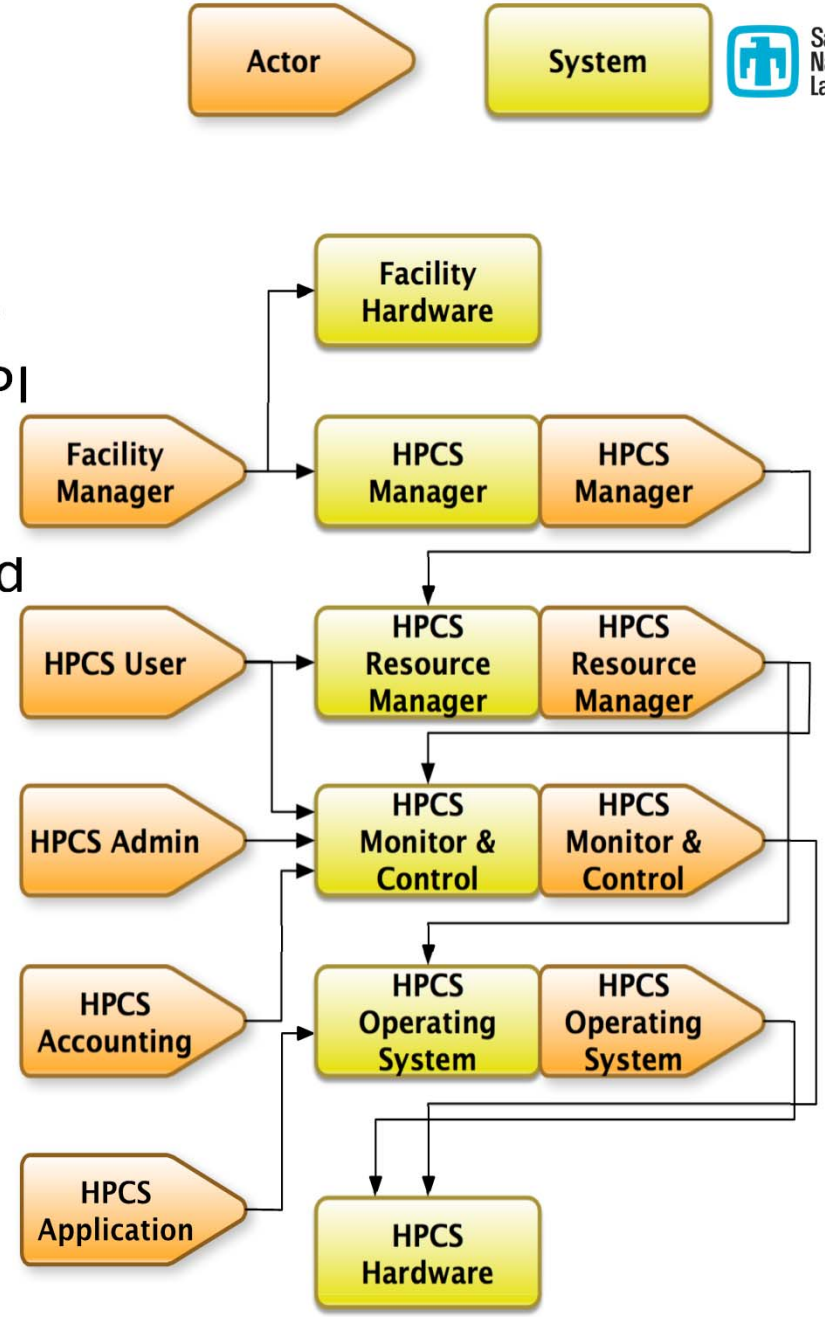
Use Case Concepts taken from UML specification ISO/IEC 19501:2005

- Use Case – A specific way of ***using*** the system by performing some part of the functionality.
- Actor – A representation of what interacts with the system. May be a person, another system, or something else (e.g. cron or async event).
- Use cases are represented by ovals. Typical naming convention is a verb followed by object. Subject is implied by the initiating actor.
- An actor is represented by a stick figure.
- There is no notion of where data resides or its layout. It's like a floating platter that one gets data off of, or puts data on.

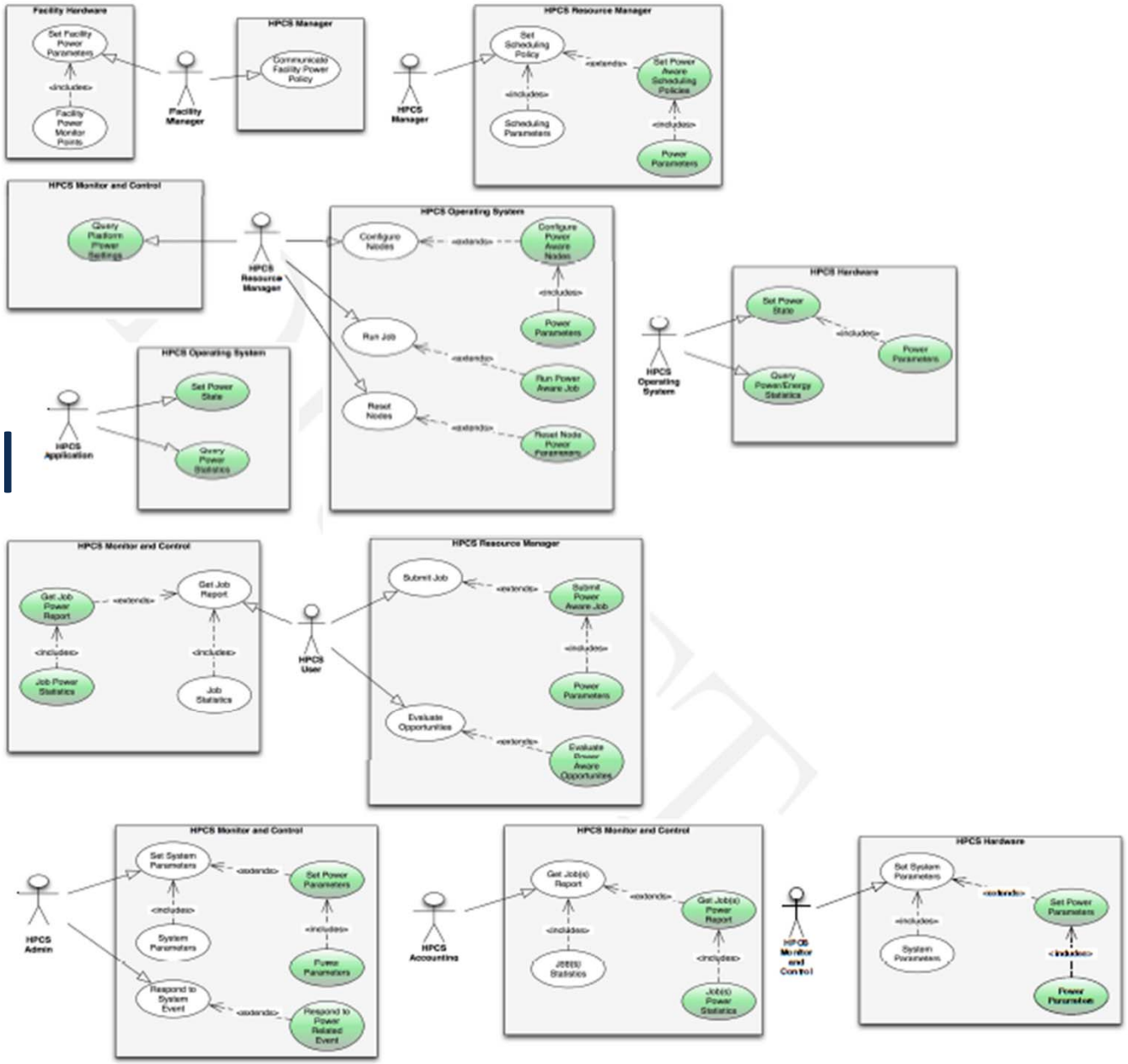


Power API Actors and Systems

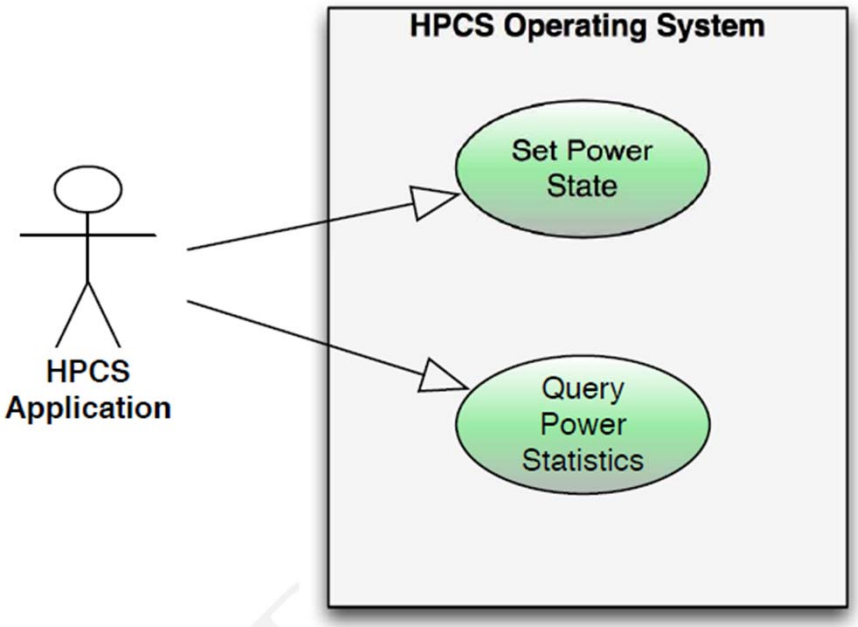
- A high level view of the entire scope to be covered by the API
- A system can also be an actor
- A Runtime System is not called out in this model. Portions of it are implemented in these Actor/Systems
  - Resource manager(s)
  - Application (Libraries)
  - Operating Systems



Our Use Case Model



Actor: HPCS Application
System: HPCS Operating System



Actor	HPCS Application
System	HPCS Operating System
Use Case	Set Power State
Description	While an application is running, it may choose to actively manage its power or energy consumption. For example, it may request to lower power state on the CPU while writing a checkpoint to disk or going into an I/O intensive phase.
Trigger	The application reaches a point in the code where it knows that the power state is either particularly important or not important.
Flow of Events	1. As a precursor to setting a new power state, the application requests information on available power states and the current power state of its components (see Query Power Statistics use case 3.9.2). 2. The operating system returns requested power/energy state and statistics. 3. If the current power state is not optimum, the application decides how much to raise or lower the power/energy state of one or more components. 4. The application requests that the power state be changed on one or more components. 5. The operating system returns success along with state information.
Alternative Paths	5a. The operating system returns failure along with state information. 5b. The Application can decide to resubmit request immediately or wait until a later point in execution.
Frequency	Could be measured in seconds, more likely in minutes
Input Data	In step 4, the application requests a power state change.
Output Data	In response to step 2, the available power states are provided in some generic, system portable fashion.
Pre Condition	Application is executing at power/energy configuration A
Post Condition	Application is executing at power/energy configuration B
Power API	yes

Status & Next Steps

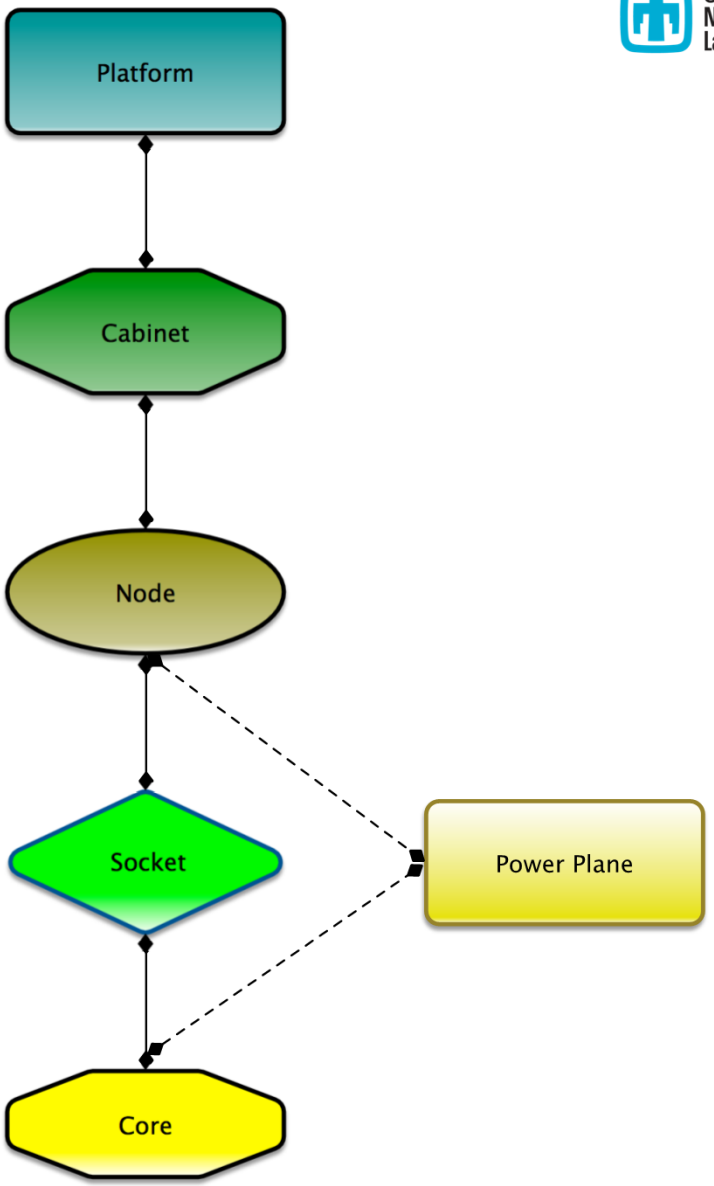
- Reviewed by Labs, Universities and Commercial partners
- Use Case document is sufficiently complete for our purposes; will release as a Sandia Report by end of calendar year
- 2014 L2 Milestone – Power API Definition/Specification

Power API Definition

- Early stages of definition/specification
- Lots of focus on foundation of API
  - What the system will look like to the user
    - What they can count on
  - What the implementer must do
    - How the implementer can extend

Base System Definition

- Aids in program portability
  - What a programmer can count on across implementations
- Guaranteed top of hierarchy object – **Platform**
  - Portable point of reference
- Set of base **DEFINED** Platform Object types that are organized in a **DEFINED** hierarchy.
  - You can count on a cabinet being below/underneath the platform object
- Set of base **DEFINED** Platform Object types that can appear anywhere in hierarchy.
  - Power Plane, for example
  - Board object another possibility
- Implementation can **EXTEND** or add Platform Object types.
  - Allow implementer as much flexibility as possible while documenting a meaningful basis for portability
- We *assume* this will evolve as we refine the API
  - For example, socket might become a defined platform object type since the goal is to support more than just CPU devices

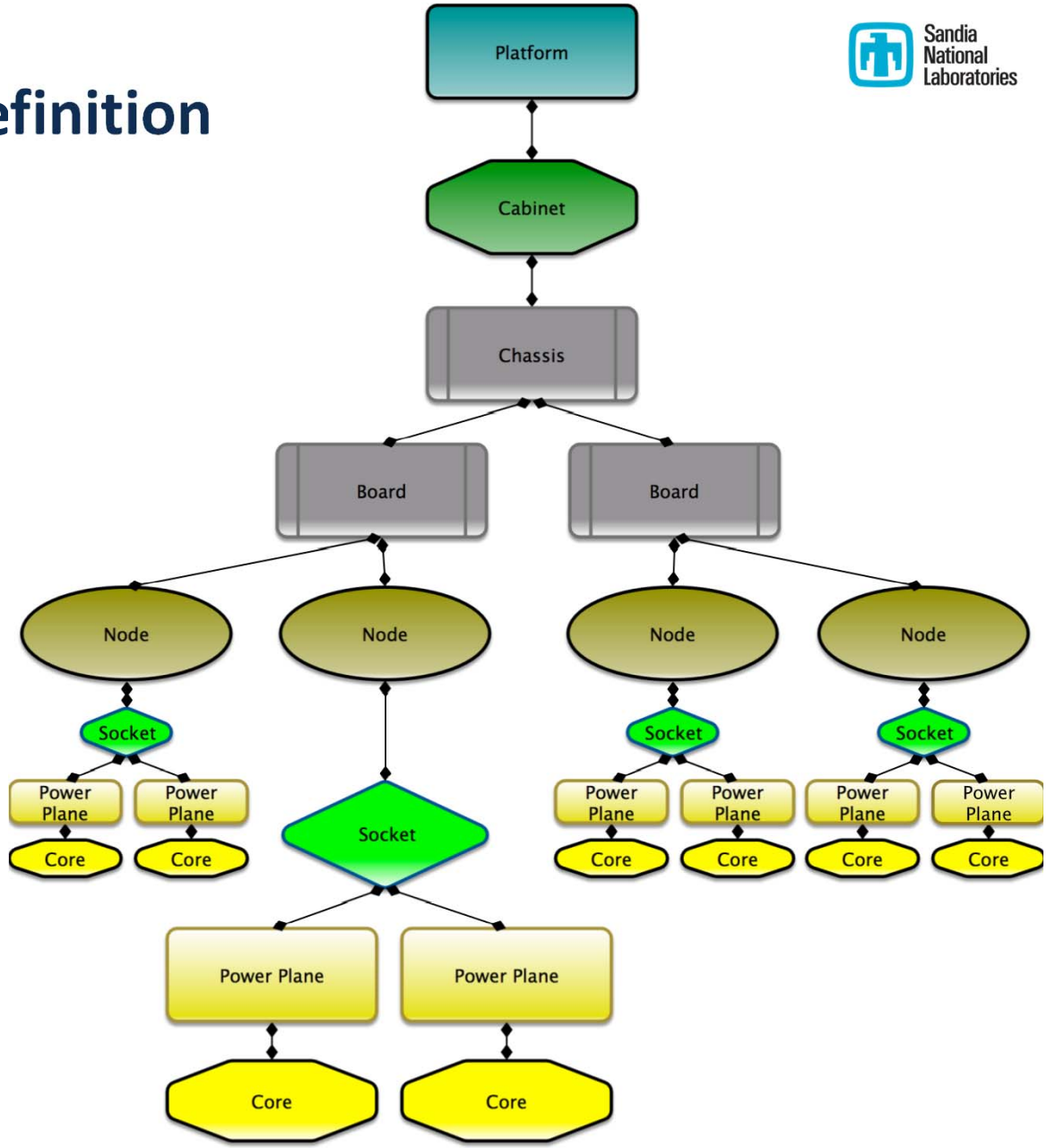


Extended System Definition

- Implementation provides an Extended System Definition based on Base System Definition
- Platform is top of hierarchy
- New Platform Object Types defined and inserted into hierarchy
  - Without violating defined order
  - Node is below Cabinet

1. Cabinets have Chassis
2. Chassis have Board(s)
3. Boards have nodes

- Extended System Definition leverages and defines separate power planes underneath Socket level to distinguish core control



Initialization

- Call to `init()` returns pointer to Platform Object type
  - Can be top of hierarchy – Platform Object – or somewhere within the extended system description (depending on where init() is called from)
- Examples
  - Called from Monitor Control -> Hardware interface `init()` returns reference to **Platform** at top of extended system description
  - Called from Operating System -> Hardware interface, `init()` required to return reference to **Node** object[3]
  - Application -> Operating System interface likewise required to return Node object [3]

[3]Node object could be a pointer to a hwloc object , for example, per http://www.open-mpi.org/projects/hwloc/

Other Features

- Navigation or Discovery calls
  - Enable navigation or traversal through system description
    - Implementation may allow discovery of entire hierarchy
    - Given the Node you *may* be allowed to discover the entire platform
- Attributes
  - Characteristics of objects
  - Some will be defined for API defined object types
  - Implementation can leverage attributes to extend capabilities
- Groups or Collections
  - A way to relate objects in logical ways
  - Again, some defined in API but can be extended by implementer

Questions…

QUESTIONS
FOUND IN GOOGLE AUTOCOMPLETE
WHY IS SEA SALT BETTER
WHY ARE THERE TREES IN THE MIDDLE OF FIELDS
WHY IS THERE NOT A POKEMON MMO
WHY IS THERE LAUGHING IN TV SHOWS
WHY ARE THERE DOORS ON THE FREEWAY
WHY ARE THERE SO MANY SVCHOST.EXE RUNNING
WHY AREN'T THERE ANY COUNTRIES IN ANTARCTICA
WHY ARE THERE SCARY SOUNDS IN MINECRAFT
WHY IS THERE KICKING IN MY STOMACH
WHY ARE THERE TWO SLASHES AFTER HTTP
WHY ARE THERE CELEBRITIES
WHY DO SNAKES EXIST
WHY DO OYSTERS HAVE PEARLS
WHY ARE DUCKS CALLED DUCKS
WHY DO THEY CALL IT THE CLAP
WHY ARE KYLE AND CARTMAN FRIENDS
WHY IS THERE AN ARROW ON AANG'S HEAD
WHY ARE TEXT MESSAGES BLUE
WHY ARE THERE MUSTACHES ON CLOTHES
WHY ARE THERE MUSTACHES ON CARS
WHY ARE THERE MUSTACHES EVERYWHERE
WHY ARE THERE SO MANY BIRDS IN OHIO
WHY IS THERE SO MUCH RAIN IN OHIO
WHY IS OHIO WEATHER SO WEIRD
WHY ARE THERE MALE AND FEMALE BIKES
WHY ARE THERE BRIDESMAIDS
WHY DO DYING PEOPLE REACH UP
WHY AREN'T THERE VARICOSE ARTERIES
WHY ARE OLD KLINGONS DIFFERENT
WHY ARE THERE SQUIRRELS
WHY IS PROGRAMMING SO HARD
WHY IS THERE A 0 OHM RESISTOR
WHY DO IGUANAS DIE
WHY AREN'T THERE DINOSAUR GHOSTS
WHY ARE THERE TINY SPIDERS IN MY HOUSE
WHY DO SPIDERS COME INSIDE
WHY ARE THERE HUGE SPIDERS IN MY HOUSE
WHY ARE THERE LOTS OF SPIDERS IN MY HOUSE
WHY ARE THERE SPIDERS IN MY ROOM
WHY ARE THERE SO MANY SPIDERS IN MY ROOM
WHY DO SPIDER BITES ITCH
WHY IS DYING SO SCARY
WHY IS THERE HELL IF GOD
WHY IS THERE NO GPS IN LAPTOPS
WHY DO KNEES CLICK
WHY AREN'T THERE E GRADES
WHY IS SEX SO IMPORTANT
WHY AREN'T ECONOMISTS RICH
WHY DO AMERICANS CALL IT SOCCER
WHY ARE MY EARS RINGING
WHY ARE THERE SO MANY AVENGERS
WHY ARE THE AVENGERS FIGHTING THE X MEN
WHY IS WOLVERINE NOT IN THE AVENGERS
WHY IS THERE LAVA
WHY ARE THERE ANTS IN MY LAPTOP
WHY IS EARTH TILTED
WHY IS SPACE BLACK
WHY IS OUTER SPACE SO COLD
WHY ARE THERE PYRAMIDS ON THE MOON
WHY IS NASA SHUTTING DOWN
WHY ARE THERE GHOSTS
WHY ARE THERE FEMALE MR
WHY IS MT VESUVIUS THERE
WHY DO THEY SAY T MINUS
WHY ARE THERE OBELISKS
WHY ARE WRESTLERS ALWAYS WET
WHY ARE OCEANS BECOMING MORE ACIDIC
WHY IS ARWEN DYING
WHY IS LIFE SO BO
WHY IS HTTPS IMPORTANT
WHY AREN'T MY ARMS GROWING
WHY ARE THERE SWARMS OF GNATS
WHY IS THERE PHLEGM
WHY ARE THERE SO MANY CROWS IN ROCHESTER, MN
WHY IS PSYCHIC WEAK TO BUG
WHY DO CHILDREN GET CANCER
WHY IS POSEIDON ANGRY WITH ODYSSEUS
WHY IS THERE ICE IN SPACE
WHY IS THERE AN OWL IN MY BACKYARD
WHY IS THERE AN OWL OUTSIDE MY WINDOW
WHY IS THERE AN OWL ON THE DOLLAR BILL
WHY DO OWLS ATTACK PEOPLE
WHY ARE AK 47s SO EXPENSIVE
WHY ARE THERE HELICOPTERS CIRCLING MY HOUSE
WHY ARE THERE GODS
WHY ARE THERE TWO SPOCKS
WHY ARE MY BOOBS ITCHY
WHY ARE CIGARETTES LEGAL
WHY ARE THERE DUCKS IN MY POOL
WHY IS JESUS WHITE
WHY IS THERE LIQUID IN MY EAR
WHY DO Q TIPS FEEL GOOD
WHY DO GOOD PEOPLE DIE
WHY AREN'T THERE GUNS IN HARRY POTTER
WHY IS THERE NO KING IN ENGL
WHY ARE DOGS AFRAID OF FIREW
WHY ARE THERE WEEKS
WHY DO I FEEL DIZZY