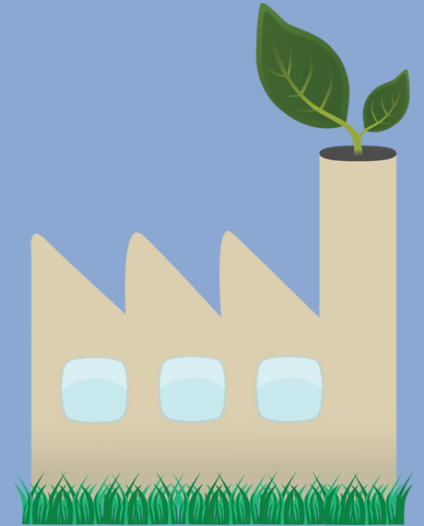
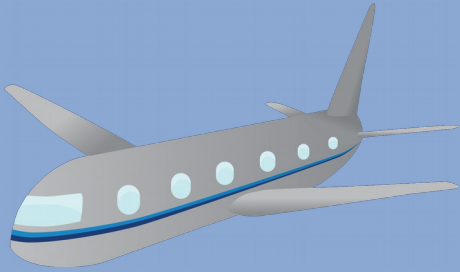


A Translationally Inspired Model for Autonomous Device Self-Regulation

Greg Wettstein, R.Ph., Ph.D
Principal Engineer, IDfusion LLC

Autonomous Self-Regulation

Brought to you by:



$$Y = mx + b$$



Contrasting the Disciplines

- **Software Assurance**

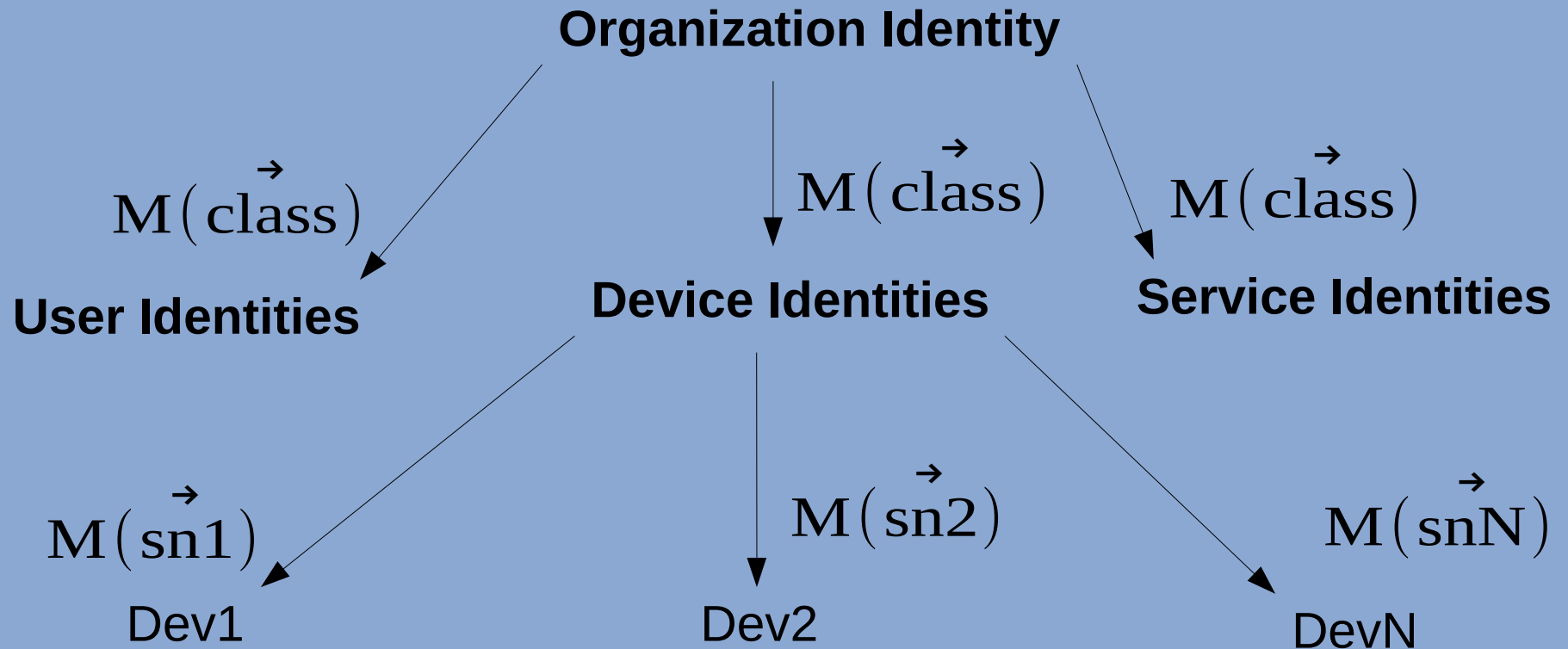
The development of a formal definition of the desired behavior of a software platform and the mathematical proof of its correctness including the binary implementation of the software.

- **Behavioral Assurance**

The development of a formal definition of the desired information exchange events for a computing platform and the mathematical verification of the hardware specific implementation of that behavior.

Our Inspiration - Identity Modeling

Genetic Hash Chaining – Identity Arborescence



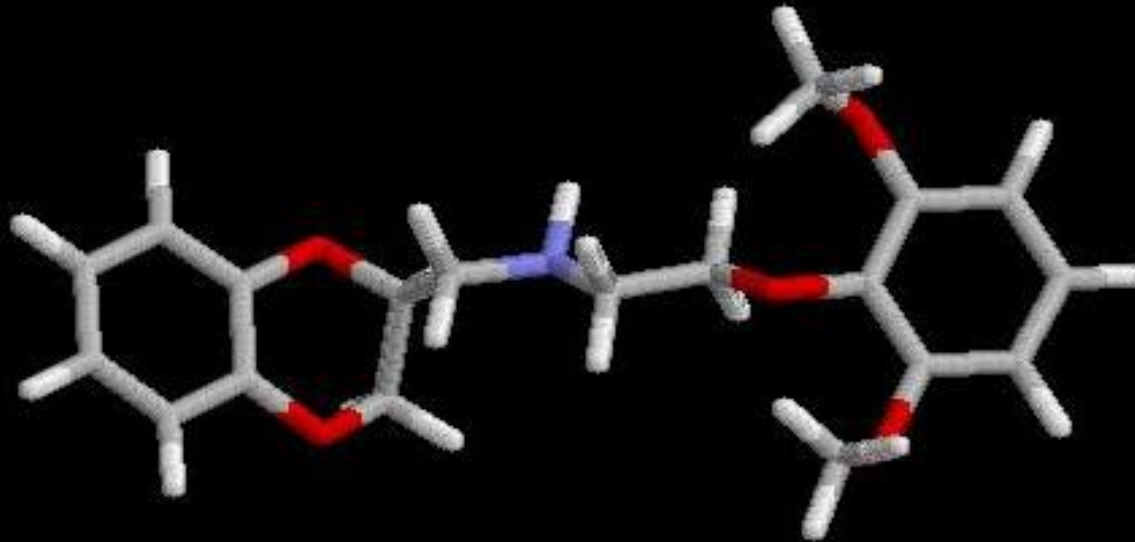
Process yields device identities which are linked by a 'measurement path' to an organizational identity.

Our Initial Motivation - NHIN

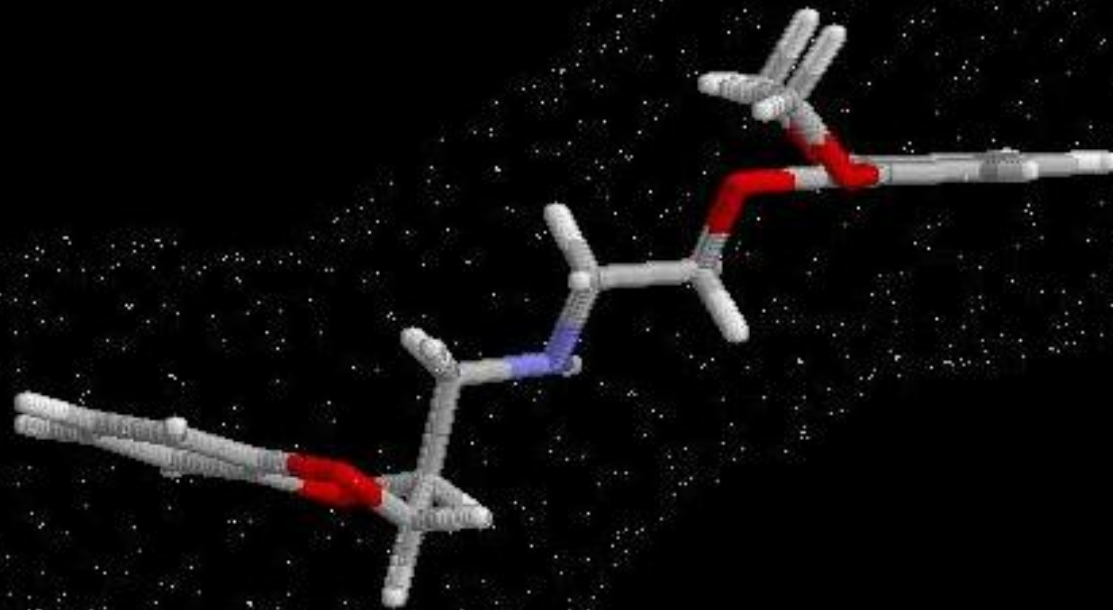
***“These efforts will coordinate patient care with
The stroke of a key or pull up life saving health
information instantly in an emergency.”***

Vice-President Biden and Secretary Sebelius
05/04/2010

Where Things Started



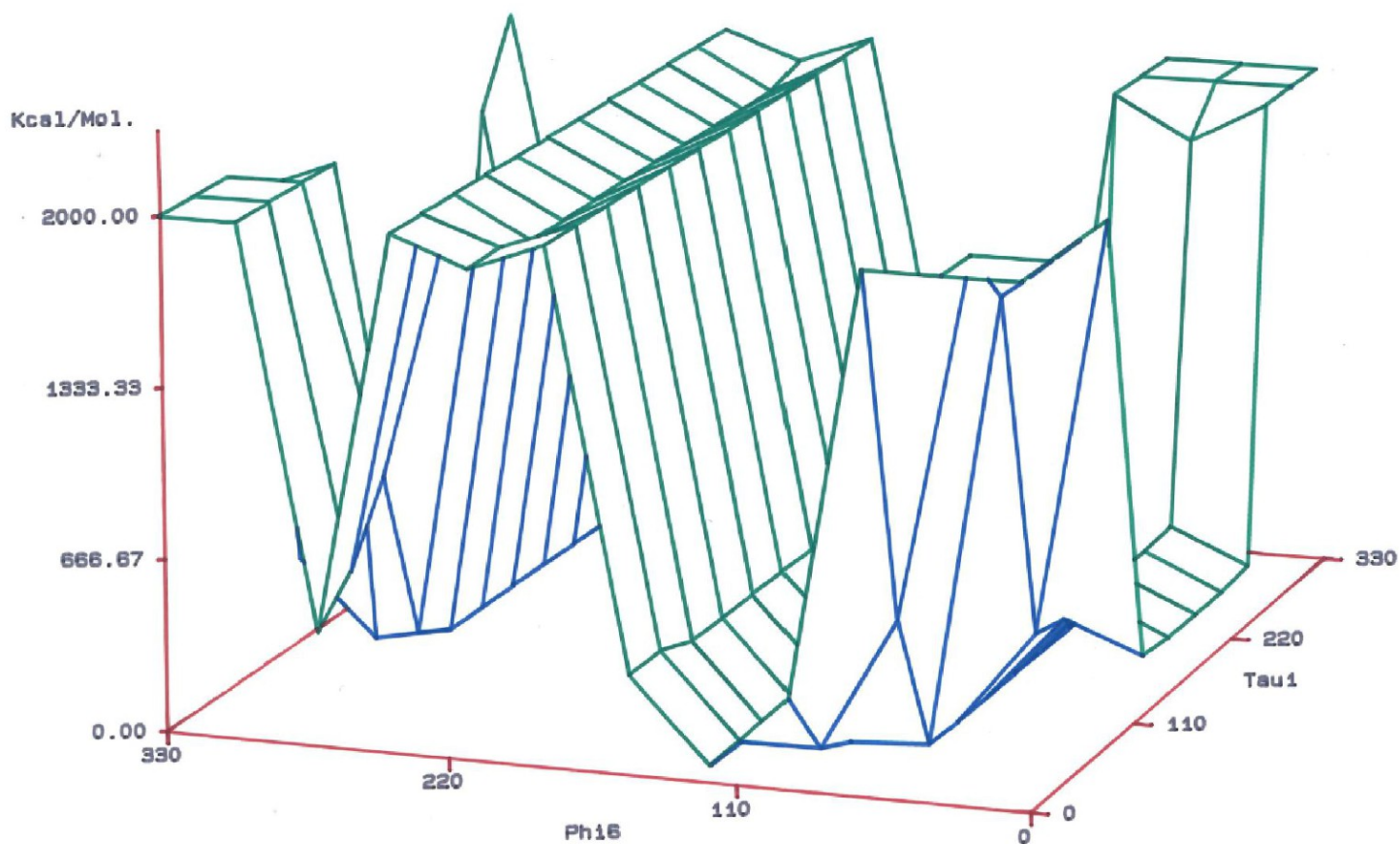
Our Objective



The Approximation



Understanding the System



158

FIGURE 8-6: Conformational energy map of Phi6-Tau1 torsional interaction. Energy relative to local minimum.

Quantum Mechanical Influences on Modeled Autonomous Systems

- Semi-empirical approximations.
- Degenerate representations of observable states.
- System trajectory paths.

Modeling Platform Behavior

Life was easy in quantum mechanics:

$$\mathcal{H}\psi = E\psi$$

Dirac-von Neumann mathematical formulation structures the 'observable' states of a system to be the result of a self-adjoint hermitian operator applied to a Hilbert space.

Commonly implemented as a Linear Combination of Atomic Orbitals using various implementations of 1-electron Gaussian functions as the basis set.

Definition of an Operating System

Formal:

The collection of software that directs a computer's operations, controlling and scheduling the execution of other programs, and managing storage, input/output, and communication resources.

From a security perspective:

A software abstraction layer which is responsible for mediating information exchange events between a context of execution (Actor) and a data source or destination (Subject).

$$B = M(A \overset{\rightarrow}{\circ} S)$$

Actor/Subject Identity Composition

Actor and subject identities are functional compressions of entity specific identity characteristics.

Actor (context of execution – process/thread)

DAC components: uid, gid, euid, eguid, suid, sguid etc.

Capability masks: effective, permitted, inheritable, ambient

Subject (inodes)

Ownership and mode.

Content digest.

Filesystem components: name, superblock.

Model parameters can be extracted by tooling from standard software development processes.

Actor/Subject Identity Modeling

- **Premise 1**

The interaction of an actor and subject identity yields an 'eigenvector' in the behavioral field.

- **Premise 2**

The individual extension sums of the behavioral eigenvectors yield the 'eigenstates' of the platform behavioral field.

- **Premise 3**

The behavior of a device is the extension sum of the device identity projected behavioral eigenstates.

$$B_{[\text{eigen}]} = \sum_{i=0}^{A_n} \sum_{j=0}^{S_{m[a]}} M(H_n(A_i) \parallel H_n(S_j))$$

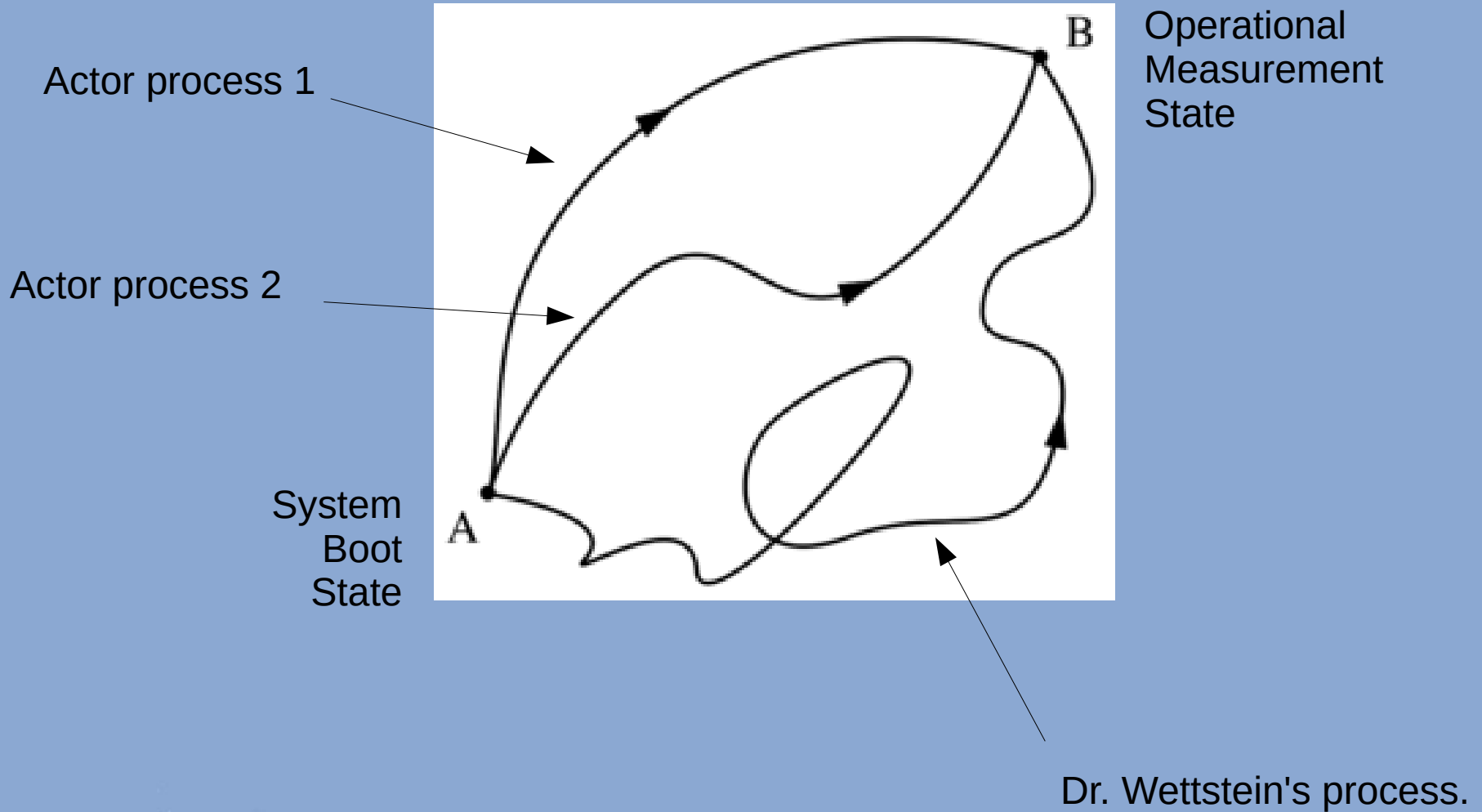
Behavior Compromise Modeling

- The Hamiltonian path of a single actor identity over its subject field yields an *'iso-identity contour'*.
 - In QM terms the *'Orbit'* of the actor through its field.
- Extra-dimensional compromise.
 - Platform behavior goes 'off-contour'.
 - Detectable by deterministic behavior modeling.
- Intra-dimensional compromise.
 - Platform behavior remains 'on-contour'.
 - Requires detection by stochastic methods.

Path Integral Formulation

With apologies to Richard Feynman

Platform Behavioral Field



Linux Implementation – 4.4 LTS

- Originally based on Linux Integrity Measure Architecture (IMA) which treated integrity only as a function of file contents for privileged users.
 - Sailer, Zhang, Jaeger, Doorn; IBM TJ Watson Research
- Extensions and eventual re-write to implement full actor/subject modeling, trajectory modeling and forensics capture.
- Significant focus on addressing engineering issues surrounding deployment tractability.

Modeling Engine Interface

/sys/kernel/security/ima/iso-identity

```
-r--r----- 1 root root 0 Aug 11 02:31 contours
-r--r----- 1 root root 0 Aug 11 02:31 forensics
--w----- 1 root root 0 Aug 11 02:31 identity
--w----- 1 root root 0 Aug 11 02:31 map
-r--r----- 1 root root 0 Aug 11 02:31 measurement
--w----- 1 root root 0 Aug 11 02:31 pseudonym
--w----- 1 root root 0 Aug 11 02:31 sealed
-r--r----- 1 root root 0 Aug 11 02:31 trajectory
```

Semi-Empirical Approximation Subject Pseudonyms

- Implemented to counter long standing problem with integrity systems secondary to writable files.
- Configured by security supervisor during system initialization process.
- Synthetic content digest is derived from platform identity.
- Digest is irrevocably lost in case of inode unlink.
- Under active development from a network inode perspective.

Degeneracy and Iso-Behavioral States

Non-commutative property of extension operator causes inclusion of time to be a symmetry breaking operation.

$$M(A_1 \overset{\rightarrow}{\circ} S_1) + M(A_2 \overset{\rightarrow}{\circ} S_1) \neq M(A_2 \overset{\rightarrow}{\circ} S_1) + M(A_1 \overset{\rightarrow}{\circ} S_1)$$

Removal of time component in a trajectory path causes the remaining behavioral eigenvectors to collapse into a set of degenerate eigenstates which are considered to be iso-behavioral representations of an observed platform state.

Model Simplification:

Load eigenvalues/contours, seal modeling engine and monitor system for extra-dimensional behaviors.

System Trajectory Paths

- The modeling engine provides a trajectory summary for the observed behavior field of the platform.
- Tooling generates behavioral eigenstates from system trajectory path as a component of the software development development process.
- Device identity projection causes each trajectory path to be unique.

```
event{swapper/0:/sbin/init} actor{uid=0, euid=0, suid=0, gid=0, egid=0, sgid=0,  
fsuid=0, fsgid=0, cap=0x3fffffff} subject{uid=0, gid=0, mode=0100700,  
name_length=10,  
name=cadb8688009d5594acdf21564a7ac45aa93d7ef7d87c303615c3f73f0ab34278,  
s_id=xvda, s_uuid=feadbeaffeadebeaffeadebeaffeadebeaf,  
digest=400ee3609683bac35d2f9d07f1646e44b10aec1d714ab4669c41623caff80541}
```

Behavioral Canisters

'A Container with a Label'

- Similar to containers (OS virtualization) of operating system resources.
 - Filesystem view (mnt), process identifiers, network resources, inter-process communications, host/domainname, user identifiers.
 - Basis for technologies such as Docker, Rkt, LXC.
- Unsharing the behavior namespace creates a new behavioral domain which models the behavior of a process and its descendants.
- Critically important for addressing deployment complexity issues.
- Current implementation uses SGX to accumulate canister specific behavior measurements.

`/sys/fs/iso-identity/update-NNNNNNNN`



Current and Future Work

- More expressive policy engine.
- Support for process scheduler intervention.
- Network socket support.
- Migration of the modeling engine to an extra-OS implementation.
 - VCHAN implementations based on Xen and seL4.
- Increasing the accessible TCB for SGX based applications.
 - Gating OCALL's on behavioral state.
 - Intel would have a great opportunity for silicon enhancements in SGXv2.

There is much work to do before we can announce our total failure to make any progress.....

Thank you to the supporting cast.

- John Grosen
- Rick Engen
- Scott Stofferahn
- Christopher Trom
- Jaci Stofferahn
- Izzy the Golden Retriever

Thank you for your indulgence.

Questions?