



Hyperproperties

Michael Clarkson and Fred B. Schneider
Cornell University

IEEE Symposium on Computer Security Foundations
June 23, 2008

Security Policies Today

- Confidentiality
- Integrity
- Availability

Formalize and verify any security policy? 

Program Correctness ca. 1970s

- Partial correctness
- Total correctness
- Mutual exclusion
- Deadlock freedom
- Starvation freedom

???

Safety and Liveness

Intuition [Lamport 1977]:

- **Safety:** “Nothing bad happens”
 - Partial correctness, mutual exclusion, access control
- **Liveness:** “Something good happens”
 - Termination, guaranteed service

Safety and Liveness

Formalization:

- **Property:** Set of (infinite) execution traces
 - Trace t satisfies property P iff $t \in P$
 - Satisfaction depends on the trace alone
 - System modeled as set of traces
- **Safety property** [Lampert 1985]:
 - Bad thing = trace prefix
- **Liveness property** [Alpern and Schneider 1985]:
 - Good thing = trace suffix

Success!

Alpern and Schneider (1985, 1987):

- **Theorem.** $(\forall P: P = \text{Safe}(P) \cap \text{Live}(P))$
- **Theorem.** *Safety proved by invariance.*
- **Theorem.** *Liveness proved by well-foundedness.*
- **Theorem.** *Topological characterization:*
 - Safety = closed sets*
 - Liveness = dense sets*

Formalize and verify any property?



Back to Security Policies

Formalize and verify any property?



Formalize and verify any security policy?



Security policy $\stackrel{?}{=}$ Property

Security Policies are not Properties

Noninterference: Commands of high users have no effect on observations of low users

- Satisfaction depends on **pairs** of traces
⇒ not a property

Average response time: Average time, over all executions, to respond to request has given bound

- Satisfaction depends on **all** traces of system
⇒ not a property

Any policy that stipulates relations among traces is not a property

→ Need satisfaction to depend on *sets* of traces

Hyperproperties

A **hyperproperty** is a set of properties

A system S **satisfies** a hyperproperty H iff $S \in H$

- A hyperproperty specifies exactly the allowed sets of traces

Hyperproperties

Security policies are hyperproperties!

- **Information flow:** Noninterference, relational noninterference, generalized noninterference, observational determinism, self-bisimilarity, probabilistic noninterference, quantitative leakage
- **Service-level agreements:** Average response time, time service factor, percentage uptime
- ...

Hyperproperties

- Safety and liveness?
- Verification?

Safety

Safety proscribes “bad things”

- A bad thing is **finitely observable** and **irremediable**
- S is a safety property [L85] iff

$$(\forall t \notin S : (\exists b \leq t : (\forall u \geq b : u \notin S)))$$

b is a finite trace

- S is a **safety hyperproperty** (“hypersafety”) iff

$$(\forall T \notin S : (\exists B \leq T : (\forall U \geq B : U \notin S)))$$

B is a finite set of finite traces

Prefix Ordering

An **observation** is a finite set of finite traces

Intuition: Observer sees a set of partial executions

$M \leq T$ (is a **prefix** of) iff:

- M is an observation, and
- $\forall m \in M : (\exists t \in T : (m \leq t))$

Intuition: If observer watched longer, M could become T

Safety Hyperproperties

- **Noninterference** [Goguen and Meseguer 1982]
 - Bad thing is a pair of traces where removing high commands does change low observations
- **Observational determinism** [Roscoe 1995]
 - Bad thing is a pair of traces that cause system to look nondeterministic to low observer

Liveness

Liveness prescribes “good things”

- A good thing is **always possible** and **possibly infinite**
- L is a liveness property [AS85] iff

$$(\forall t : (\exists g \geq t : g \in L))$$

t is a finite trace

- L is a **liveness hyperproperty** (“hyperliveness”) iff

$$(\forall T : (\exists G \geq T : G \in L))$$

T is a finite set of finite traces

Liveness Hyperproperties

- **Average response time**
 - Good thing is that average time is low enough
- **Generalized noninterference** [McCullough 1987]
 - Good thing is additional interleavings of traces

Possibilistic Information Flow

PIF policies can be expressed with *closure operators*
[Mantel 2000]

Theorem. *All PIF policies are hyperliveness.*

Relating Properties and Hyperproperties

Can **lift** property T to hyperproperty $[T]$

- Satisfaction is equivalent iff $[T] = \mathcal{P}(T)$
- **Theorem.** S is safety $\Rightarrow [S]$ is hypersafety.
- **Theorem.** L is liveness $\Rightarrow [L]$ is hyperliveness.
- **Theorem.** *Hypersafety = closed sets.*
- **Theorem.** *Hyperliveness = dense sets.*

Safety and Liveness is a Basis

Theorem. $(\forall H : H = \text{Safe}(H) \cap \text{Live}(H))$

Probabilistic Hyperproperties

To incorporate probability:

- Assume probability on state transitions
- Construct probability measure on traces [Halpern 2003]
- Use measure to express hyperproperties

We've expressed:

- Probabilistic noninterference
- Quantitative leakage
- Channel capacity

Beyond Hyperproperties?

Add another level of sets?

Theorem. *Set of hyperproperties \equiv hyperproperty*

→ Hyperproperties are expressively complete
(for systems and trace semantics)

By analogy to logic:

- Adding levels of sets = increasing the order of logic
 - Properties = first-order predicates on traces
 - Hyperproperties = second-order
- Higher-order logic reducible to second-order

Stepping Back...

- Safety and liveness? ✓
- Verification?

Verification of Information Flow

- Barthe, D'Argenio, and Rezk (2004):
 - Reduce noninterference to a property with *self-composition*
- Terauchi and Aiken (2005):
 - Generalize to verify any *2-safety property*
 - “Property that can be refuted by observing two finite traces”

Methodology:

- Transform system to reduce 2-safety to safety property
- Verify safety property

k -Safety Hyperproperties

A **k -safety hyperproperty** is a safety hyperproperty in which the bad thing never has more than k traces

$$(\forall T \notin S : (\exists B \leq T : |B| \leq k \wedge (\forall U \geq B : B \notin S)))$$

Examples:

- **1-hypersafety:** the lifted safety properties
- **2-hypersafety:** Terauchi and Aiken's 2-safety properties
- **k -hypersafety:** $SEC(k)$ = "System can't, across all runs, output all shares of a k -secret sharing"
- **Not k -hypersafety for any k :** $SEC = \bigcup_k SEC(k)$

Verifying k -Hypersafety

Theorem. *Any k -safety hyperproperty of S is equivalent to a safety property of S^k .*

- Yields methodology for k -hypersafety
 - Incomplete for hypersafety

Logic and Verification

Full second-order logic cannot be effectively and completely axiomatized

But fragments can be...

- Might suffice for security policies

Refinement Revisited

Stepwise refinement:

- Development methodology for properties
- Uses refinement of nondeterminism
 - Satisfaction of properties is **refinement-closed**
 - But not of hyperproperties, in general

Theorem. *All safety hyperproperties are refinement-closed.*

- Refinement applicable to hypersafety
- But not all hyperproperties (necessarily)

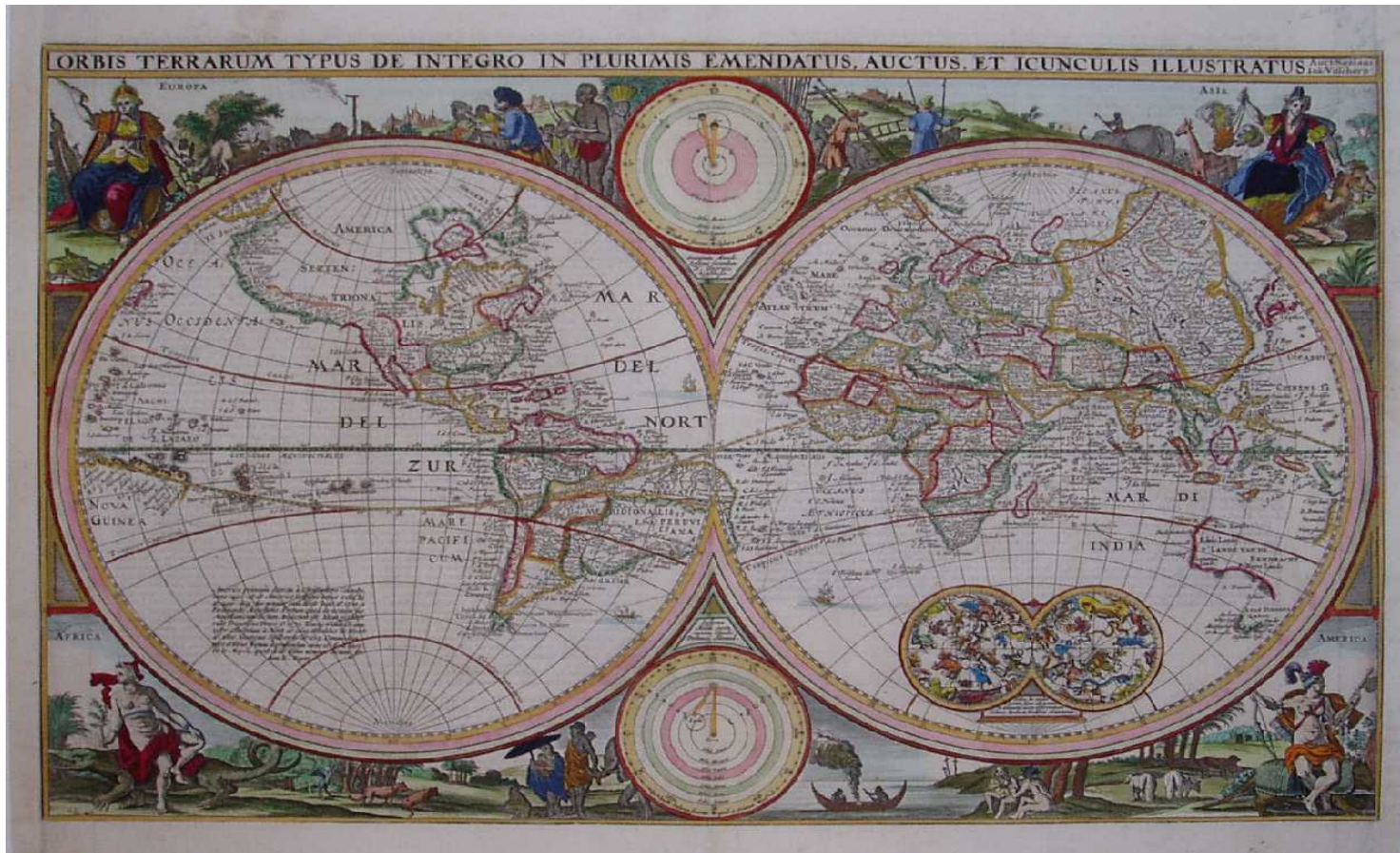
Summary

We developed a theory of hyperproperties

- Parallels theory of properties
 - Safety, liveness (basis)
 - Verification (for k -hypersafety)
 - Refinement (hypersafety)
- Expressive completeness

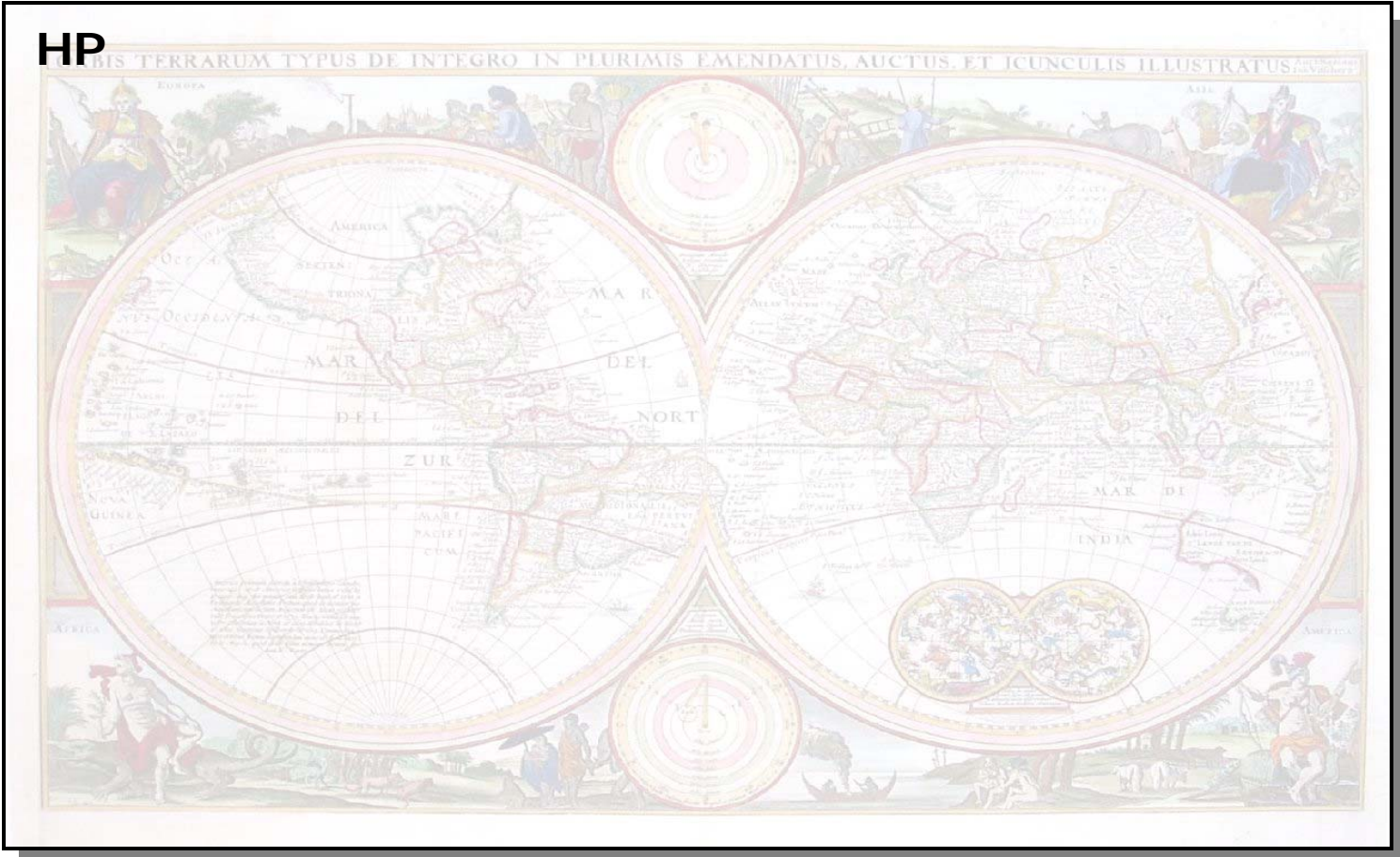
〔Currently verifying proofs using Isabelle/HOL with Denis Bueno (Cornell, Sandia)〕

Enables classification of security policies...

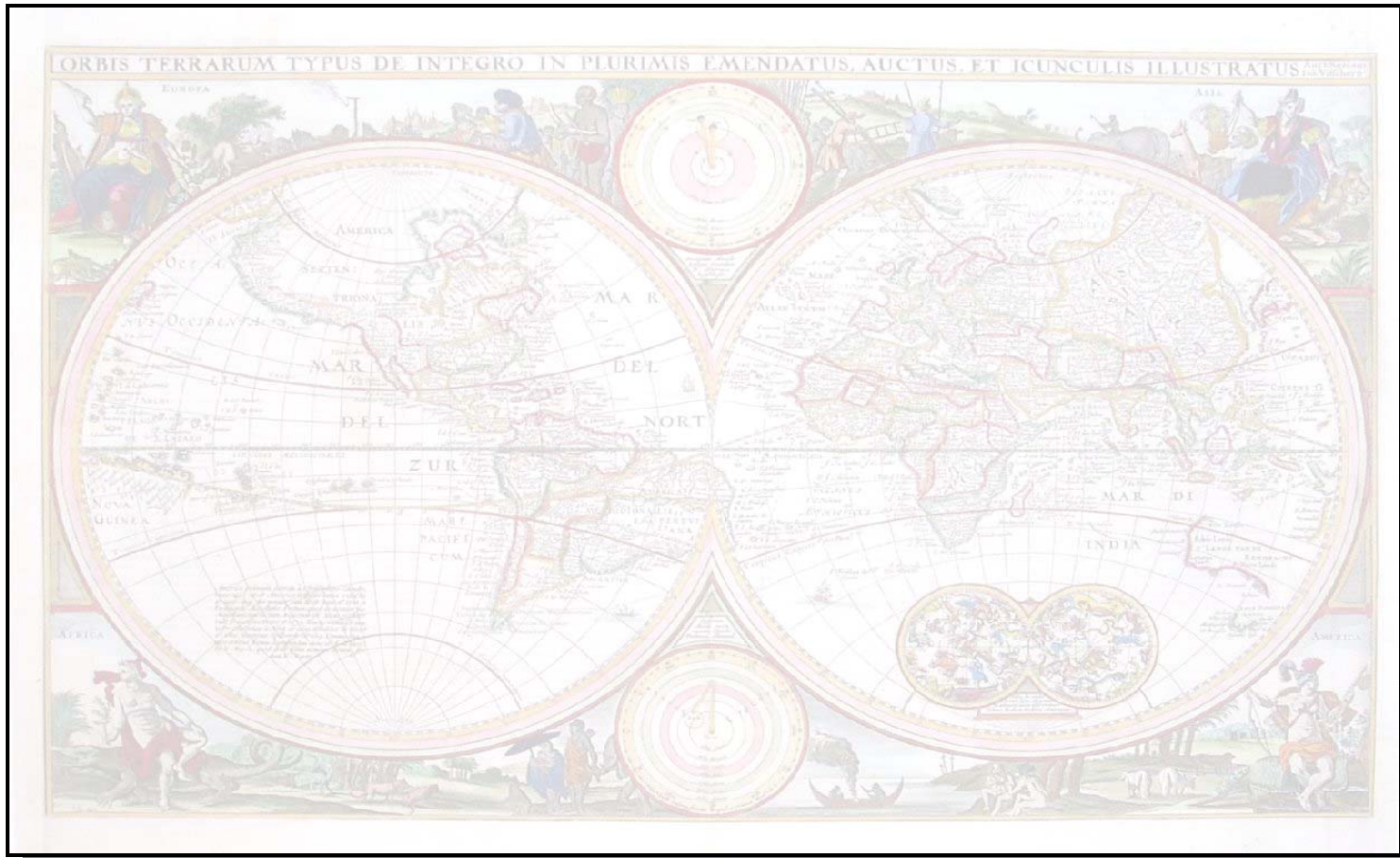


Charting the landscape...

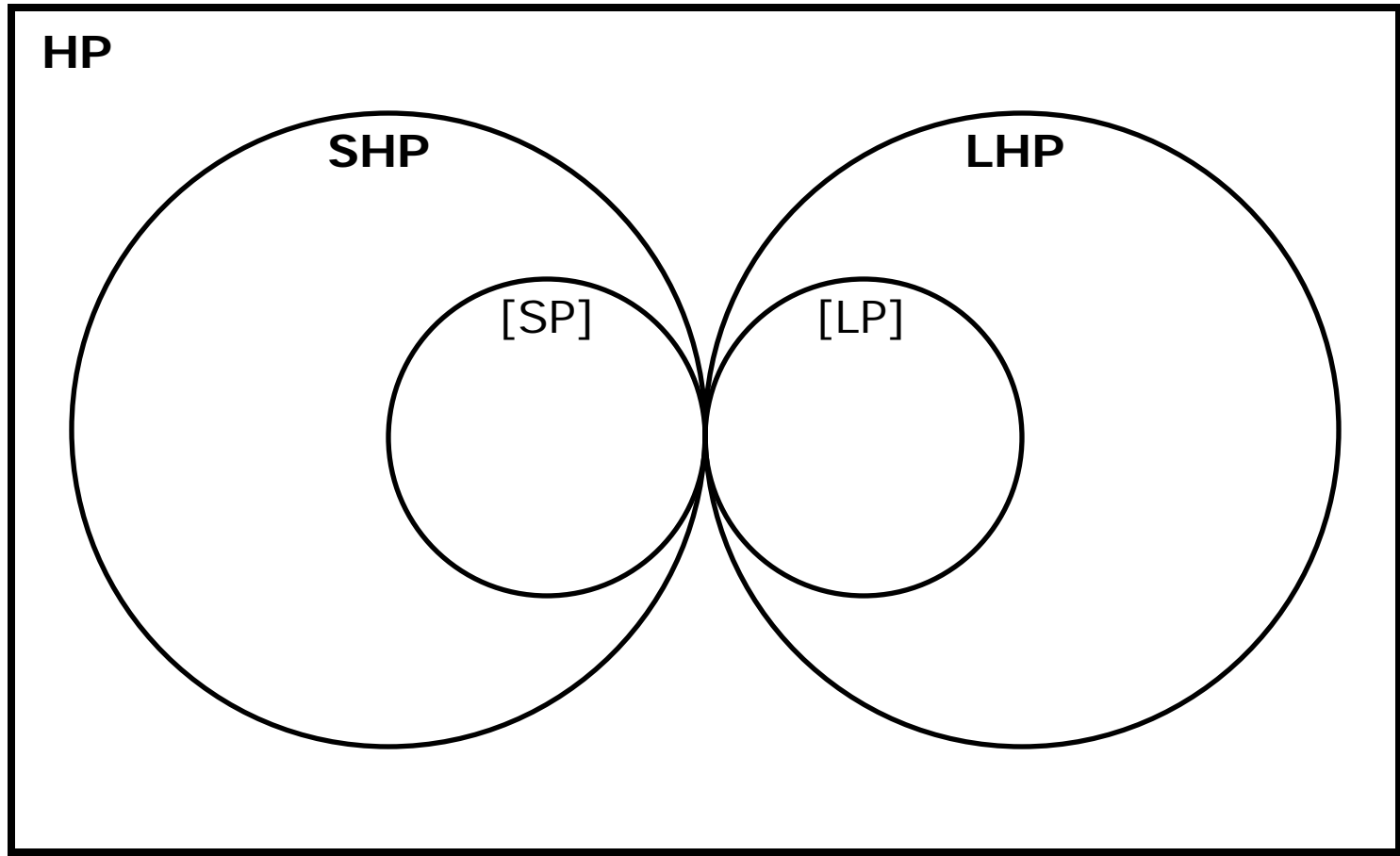
HP



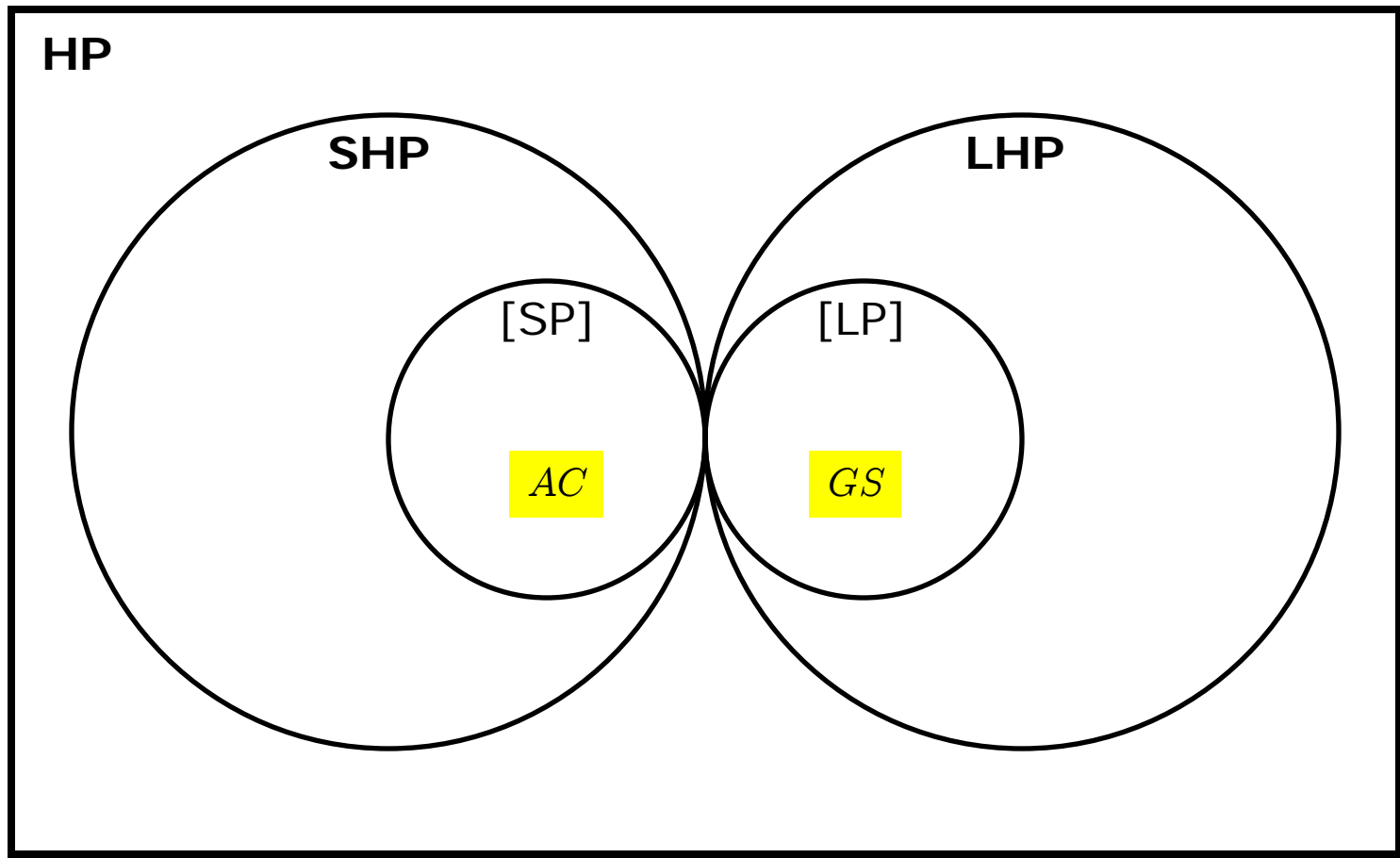
All hyperproperties (HP)



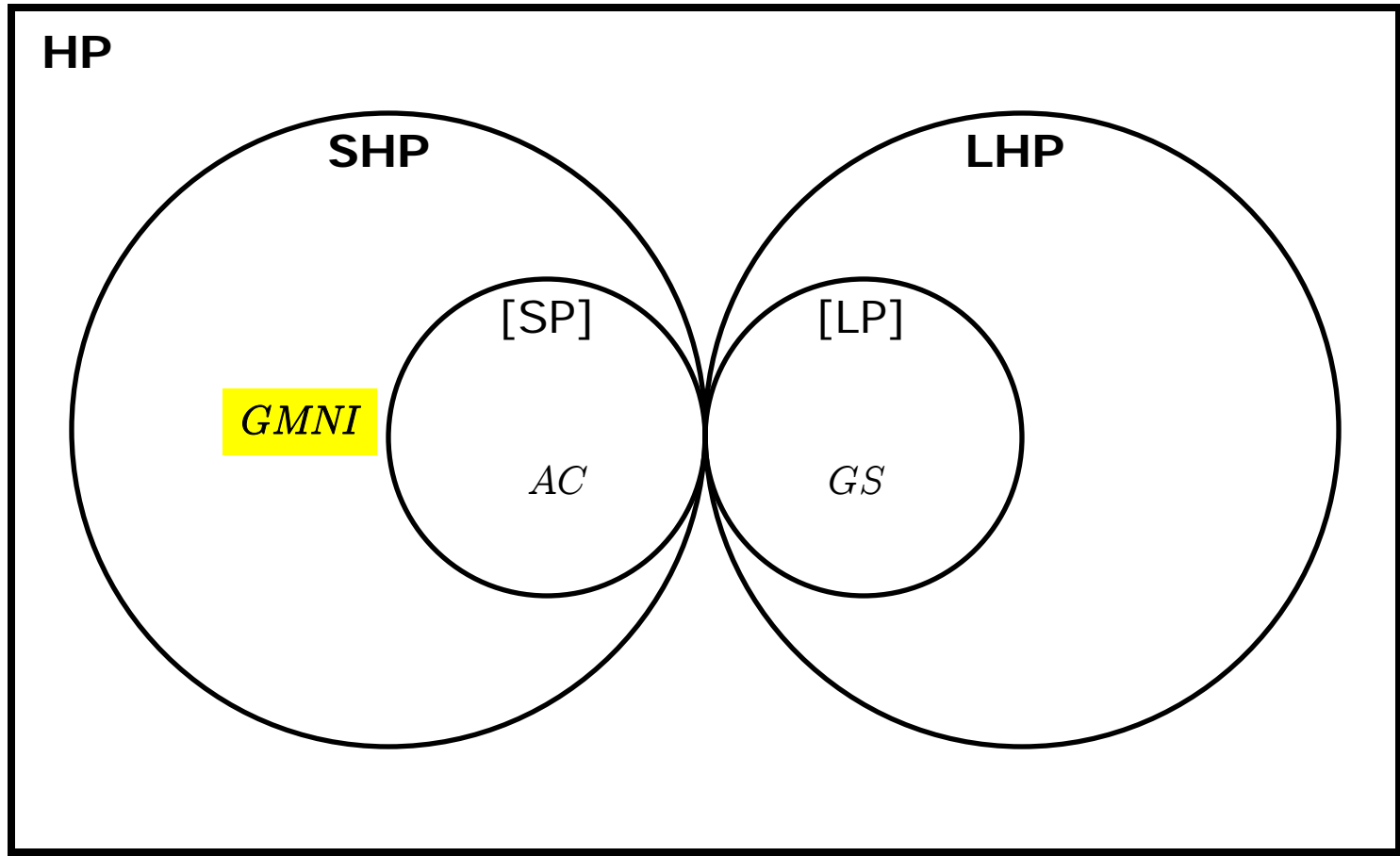
Safety hyperproperties (**SHP**)
Liveness hyperproperties (**LHP**)



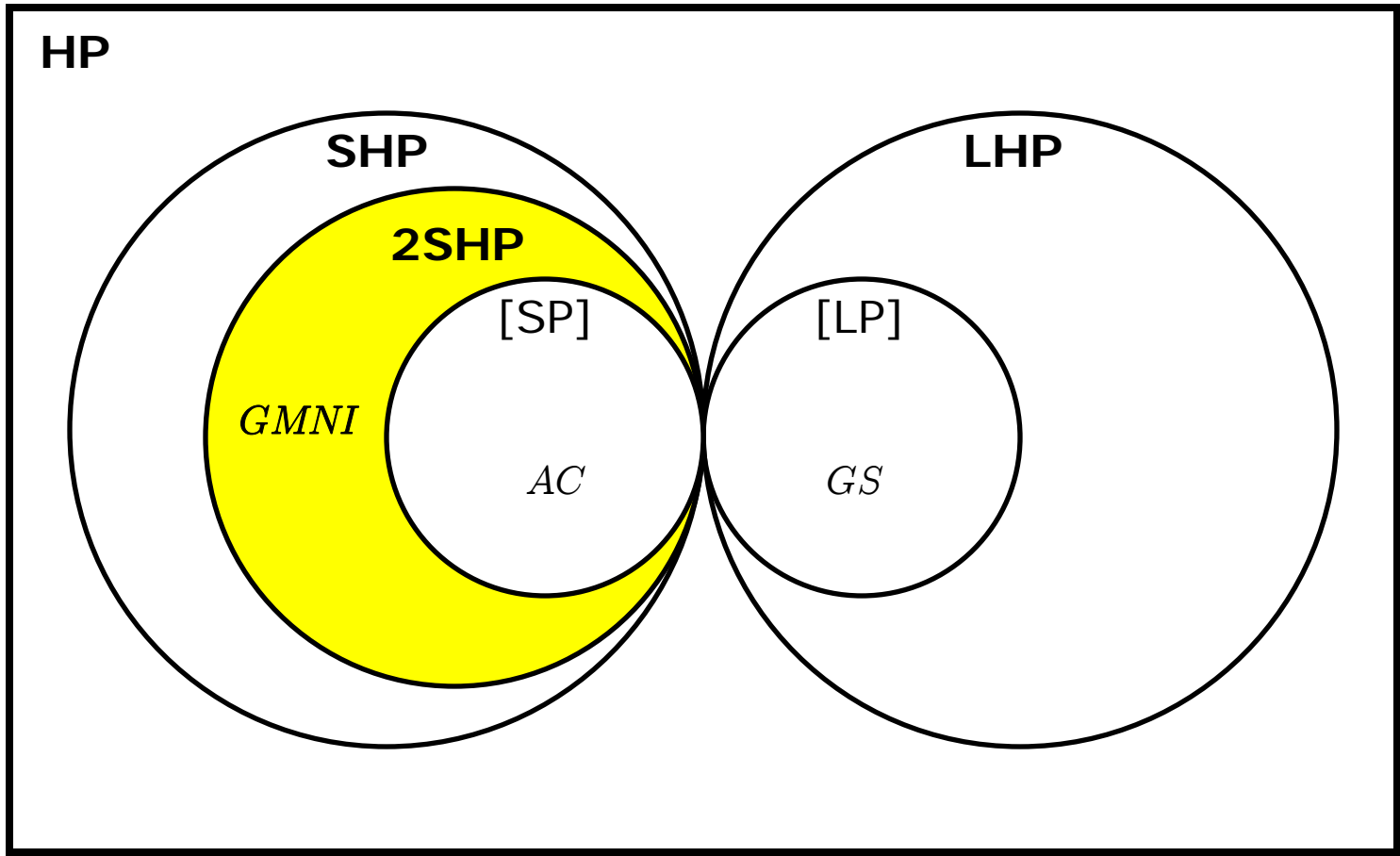
Lifted safety properties [SP]
Lifted liveness properties [LP]



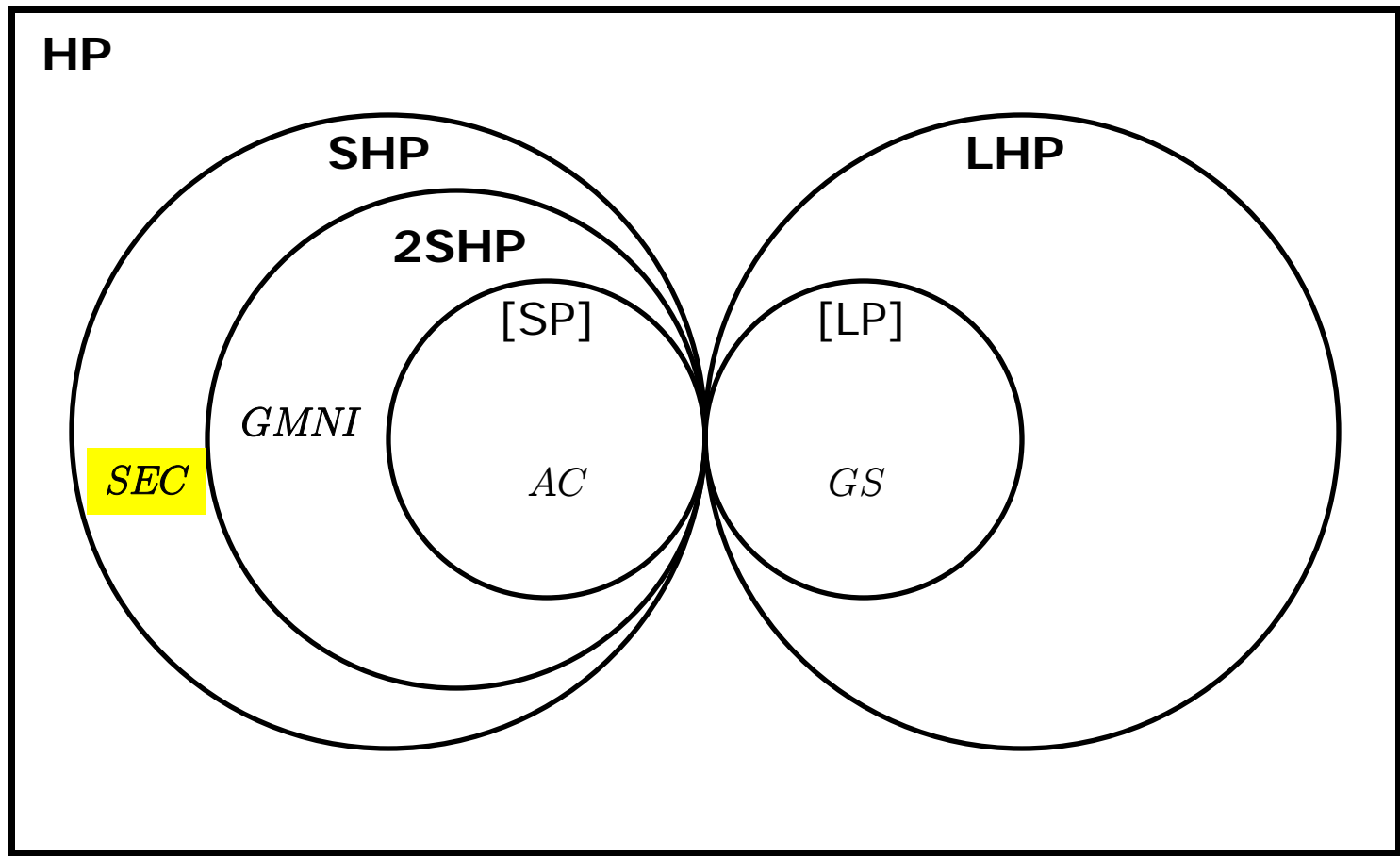
Access control (AC) is safety
Guaranteed service (GS) is liveness



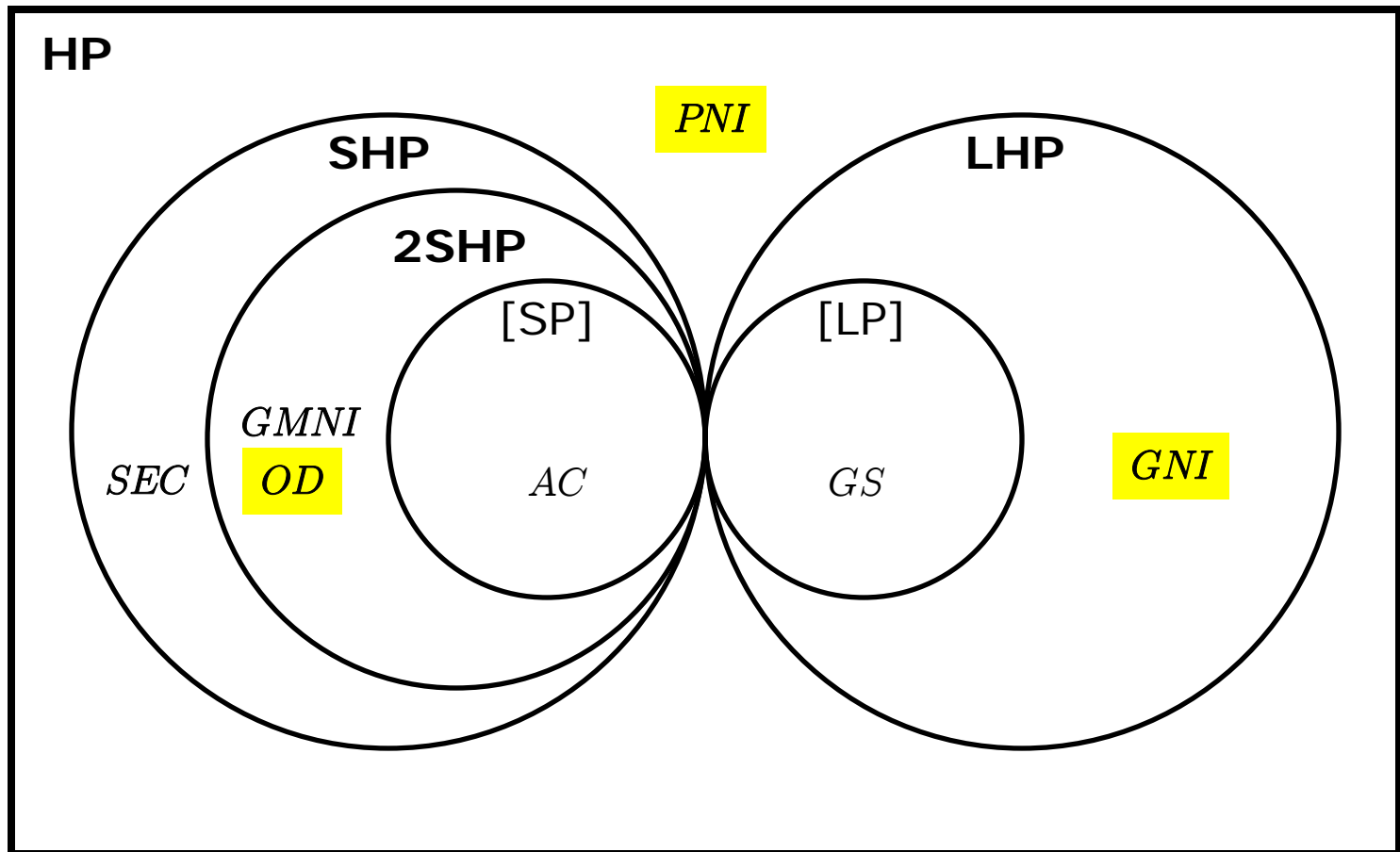
Goguen and Meseguer's noninterference (**GMNI**)
is 2-hypersafety



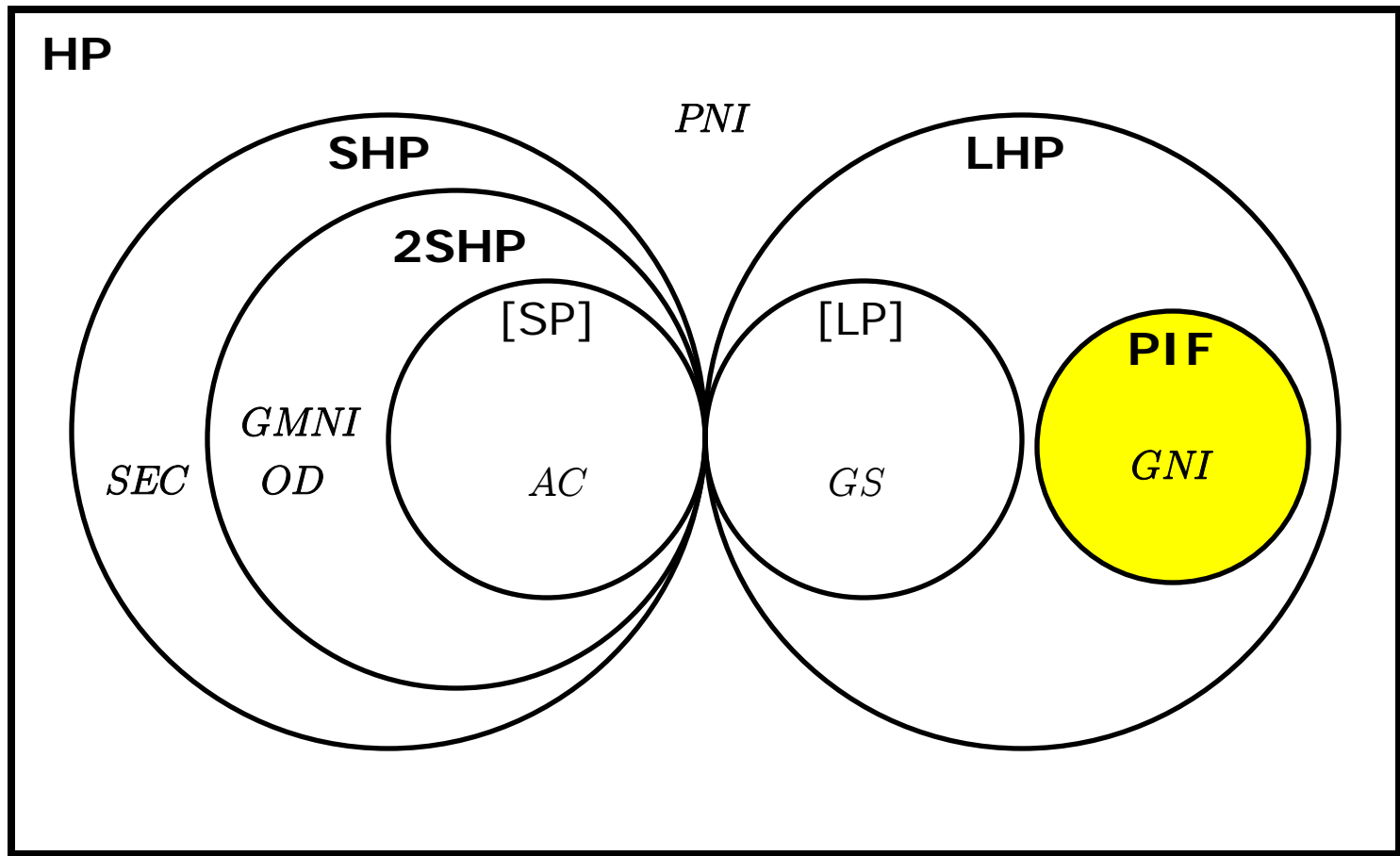
2-safety hyperproperties (**2SHP**)



Secret sharing (*SEC*) is not k -hypersafety for any k



Observational determinism (*OD*) is 2-hypersafety
 Generalized noninterference (*GNI*) is hyperliveness
 Probabilistic noninterference (*PNI*) is neither



Possibilistic information flow (**PIF**) is hyperliveness

Revisiting the CIA Landscape

○ Confidentiality

- Information flow is not a property
- Is a hyperproperty (HS: *OD*; HL: *GNI*)

○ Integrity

- Safety property?
- Dual to confidentiality, thus hyperproperty?

○ Availability

- Sometimes a property (max. response time)
- Sometimes a hyperproperty (HS: % uptime, HL: avg. resp. time)

→ CIA seems orthogonal to hyperproperties



Hyperproperties

Michael Clarkson and Fred B. Schneider
Cornell University

IEEE Symposium on Computer Security Foundations
June 23, 2008

Extra Slides

Noninterference is not a Property

- Suppose NI is a property
 - System T (for *true*) should satisfy NI
 - $L:=H$ refines T
 - And shouldn't satisfy NI
 - But since satisfaction closed under refinement,
 - $L:=H$ should satisfy NI
- Contradiction!
- Therefore, NI is not a property

Information Flow Hyperproperties

- **Noninterference:** The set of all properties T where for each trace $t \in T$, there exists another trace $u \in T$, such that u contains no high commands, but yields the same low observation as t .
- **Generalized noninterference:** The set of all properties T where for any traces t and $u \in T$, there exists a trace $v \in T$, such that v is an interleaving of the high inputs from t and the low events from u .
- **Observational determinism:** The set of all properties T where for all traces t and $u \in T$, and for all $j \in \mathbb{N}$, if t and u have the same first $j-1$ low events, then they have equivalent j^{th} low events.
- **Self-bisimilarity:** The set of all properties T where T represents a labeled transition system S , and for all low-equivalent initial memories m_1 and m_2 , the execution of S starting from m_1 is bisimilar to the execution of S starting from m_2 .

Topological Characterization

Theorem. *Our topology is equivalent to the lower Vietoris construction applied to the Plotkin topology.*

Powerdomains

- We use the *lower (Hoare) powerdomain*
 - Our \leq is the Hoare order
 - Lower Vietoris = lower powerdomain [Smyth 1983]
- Other powerdomains?
 - Change the notion of “observable”
 - Upper: Observations can disappear
 - Convex: Can observe impossibility of production of state
 - But might be useful on other semantic domains

Future Work

- Verification methodology
 - Hyperliveness?
 - Axiomatizable fragments of second order logic?
- CIA: Express with hyperproperties?
- Hyperproperties in other semantic domains