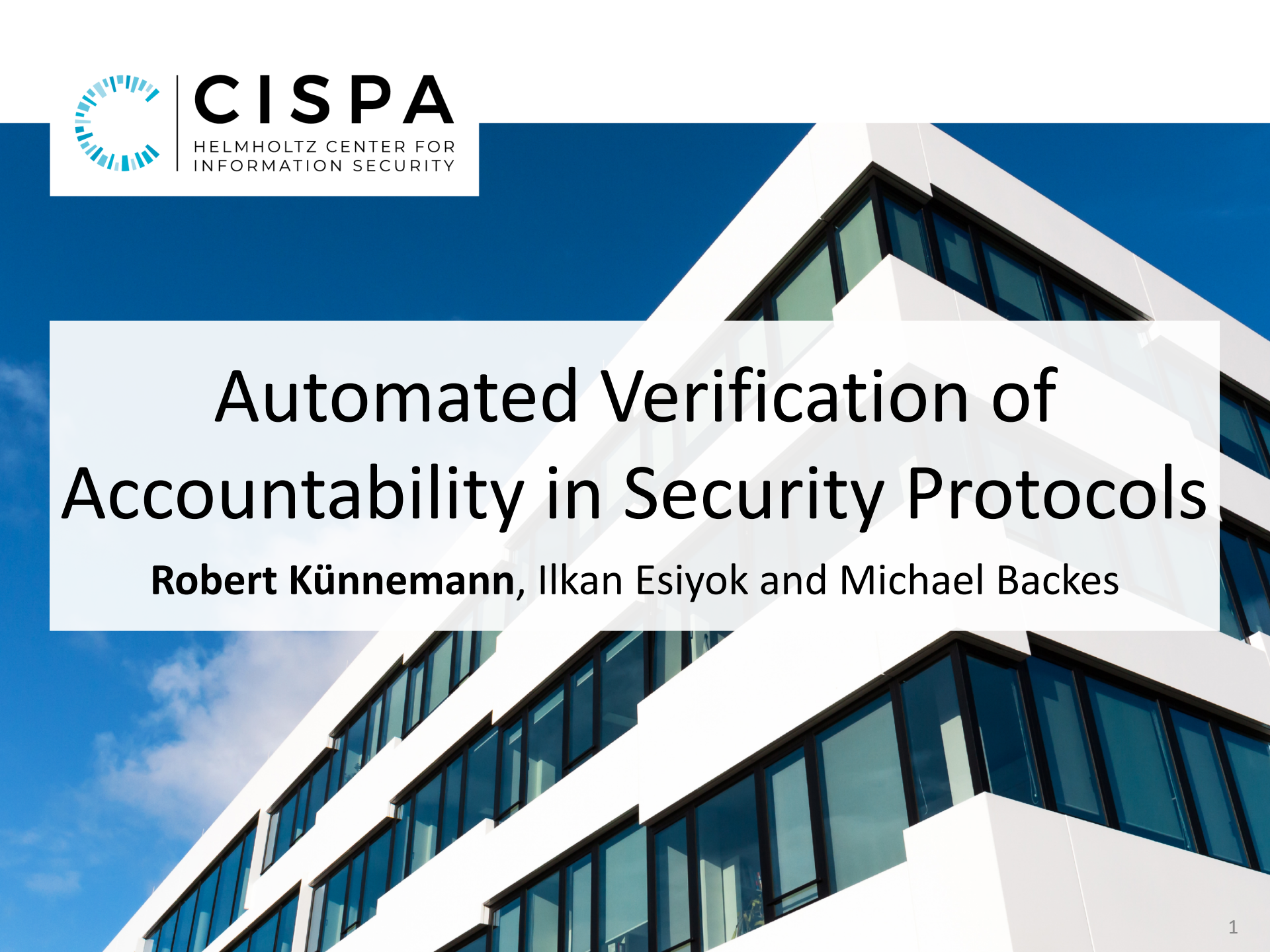
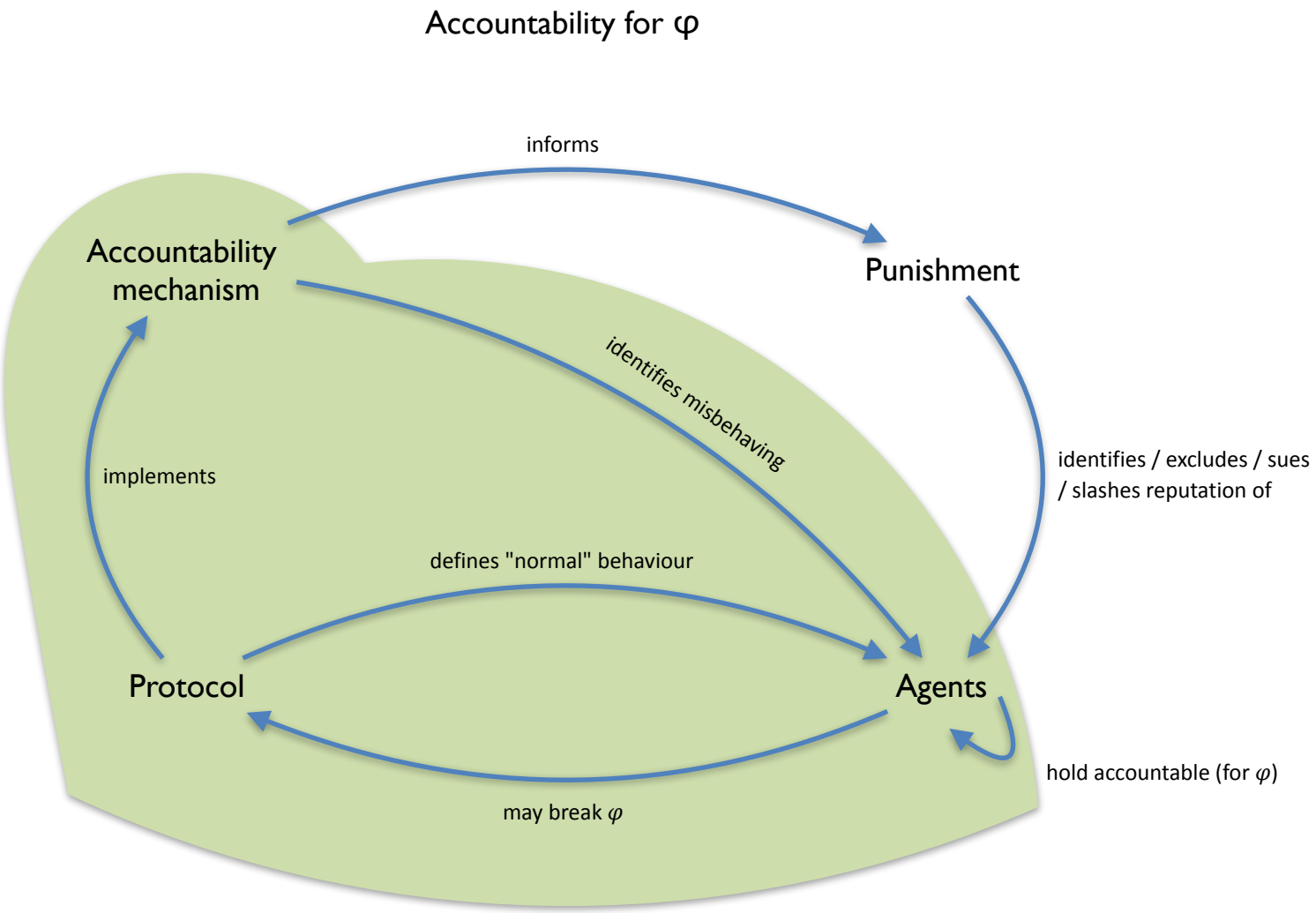


Automated Verification of Accountability in Security Protocols

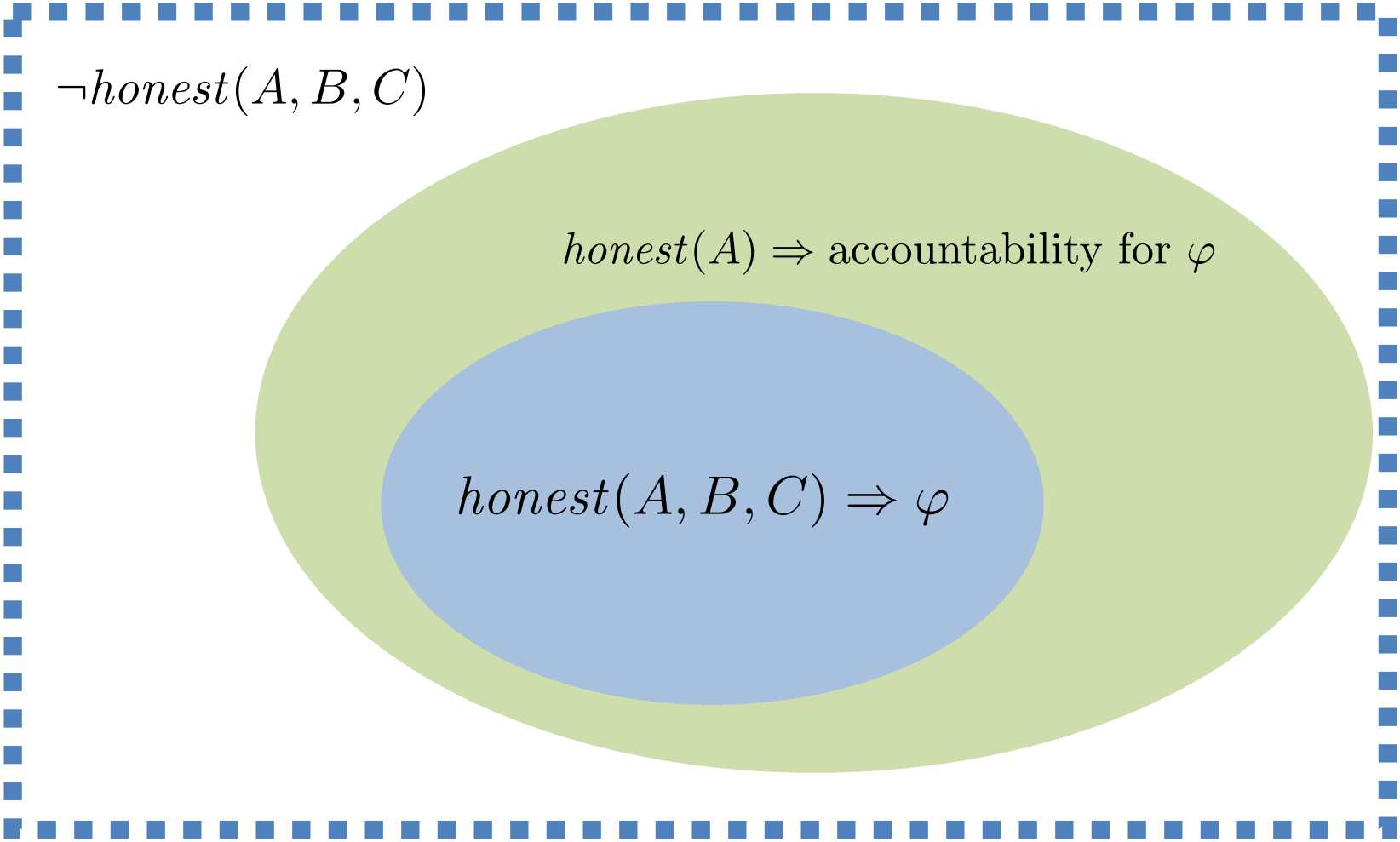
Robert Künnemann, Ilkan Esiyok and Michael Backes

Part I: What we talk about when we talk about accountability

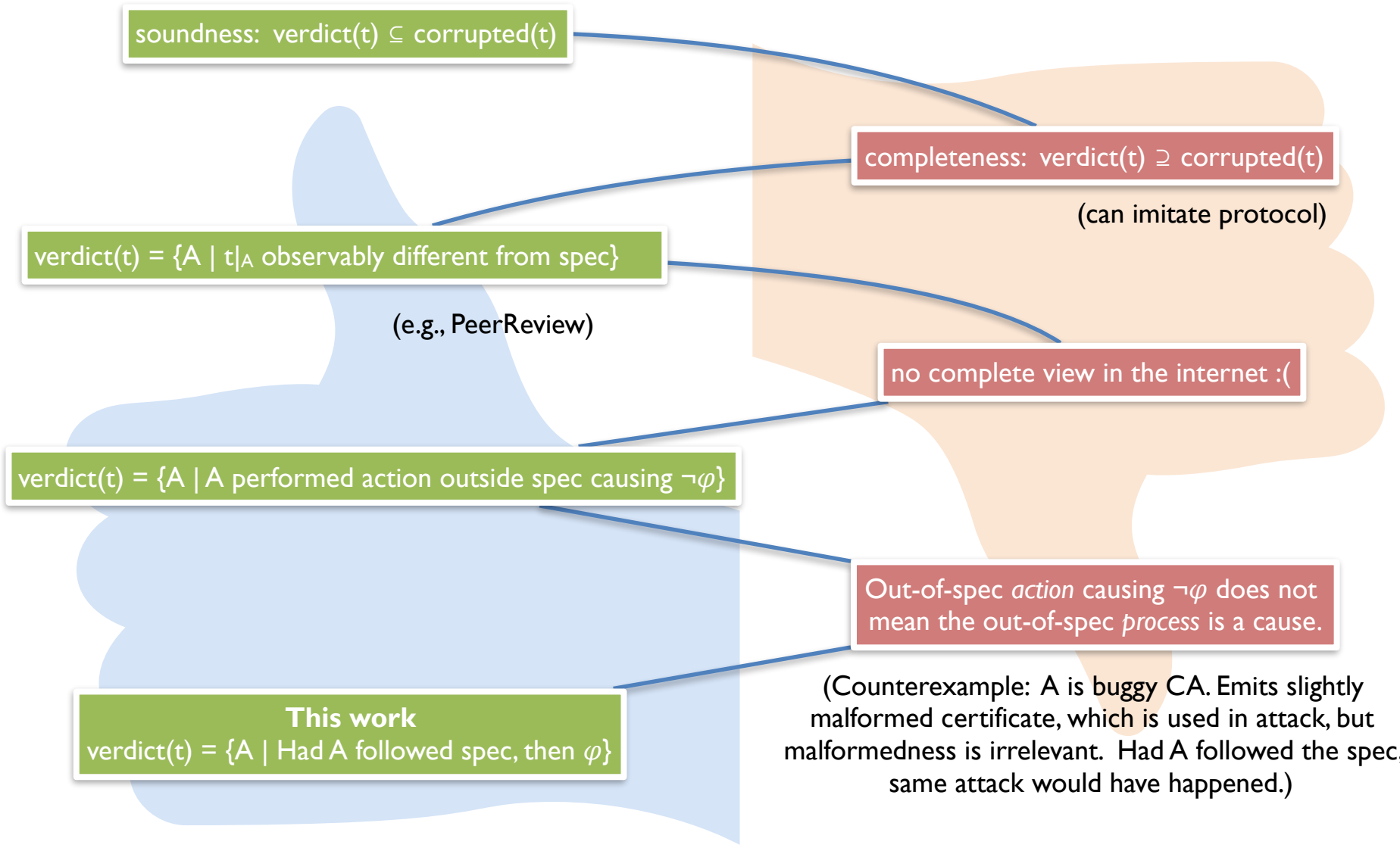
Ontology


Accountability for φ
informs
Accountability mechanism
Punishment
identifies misbehaving
implements
identifies / excludes / sues / slashes reputation of
defines "normal" behaviour
Protocol
Agents
hold accountable (for φ)
may break φ

Why Accountability



Why is it so hard?



Causation

- Event(s) A caused $\neg\varphi$ iff
  - A and $\neg\varphi$, in fact, happened.
  - In any counterfactual where A happens, $\neg\varphi$ happens.
  - A is subset-minimal.

- "Umbrella" caused "not wet", as
  - I had an umbrella and did not get wet.
  - As long as I have my umbrella, I cannot get wet.
  - Without the umbrella, I could get wet.

Causation

- Event(s) A caused $\neg\varphi$ iff
  - A and $\neg\varphi$, in fact, happened.
  - In any counterfactual where A happens, $\neg\varphi$ happens.
  - A is subset-minimal.

- Output all sets of parties S, s.t.
  - $t \models \neg\varphi$ and corrupted(t)$\supseteq$S
  - there is related t' s.t. $t' \models \neg\varphi$ and corrupt(t')=S,
  - S is subset-minimal.

Part II: Accountability in terms of trace properties

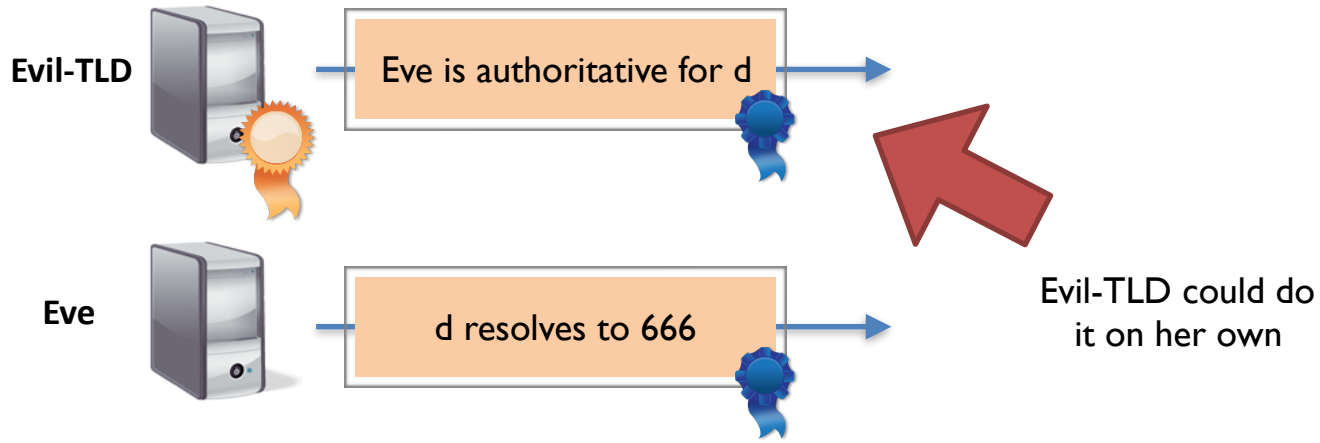
Case 1: weakest possible relation

- Consider t' is related to t iff corrupt(t') ⊆ corrupt(t)
- Idea: verdict function defined as

$$verdict(t) = \begin{cases} V_1 & if\ \omega_1(t) \\ \vdots & \\ V_n & if\ \omega_n(t) \end{cases}$$

- cases are **exhaustive** and **exclusive**, and for each i:
- **sufficiency**: Agents in V_i can produce violating trace
- **verifiability**: $V_i = \varnothing \iff \varphi$
- **minimality**: can't do with less than $S \in V_i$
- **uniqueness**: whenever ω_i is observed, parties in V_i are corrupted
- **completeness**: (omitted)

Case 2: arbitrary relation



- "But that's not what happened" -> relation r between t and t'
- idea for translation: cases are liftings R of relation r
- combination of 11 different conditions, including lifting condition:

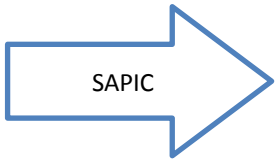
$$verdict(t) = \begin{cases} V_1 & \text{if } \omega_1(t) \\ V_2 & \text{if } \omega_2(t) \\ V_3 & \text{if } \omega_3(t) \end{cases}$$

R

Part III: Implementation

Part III Implementation

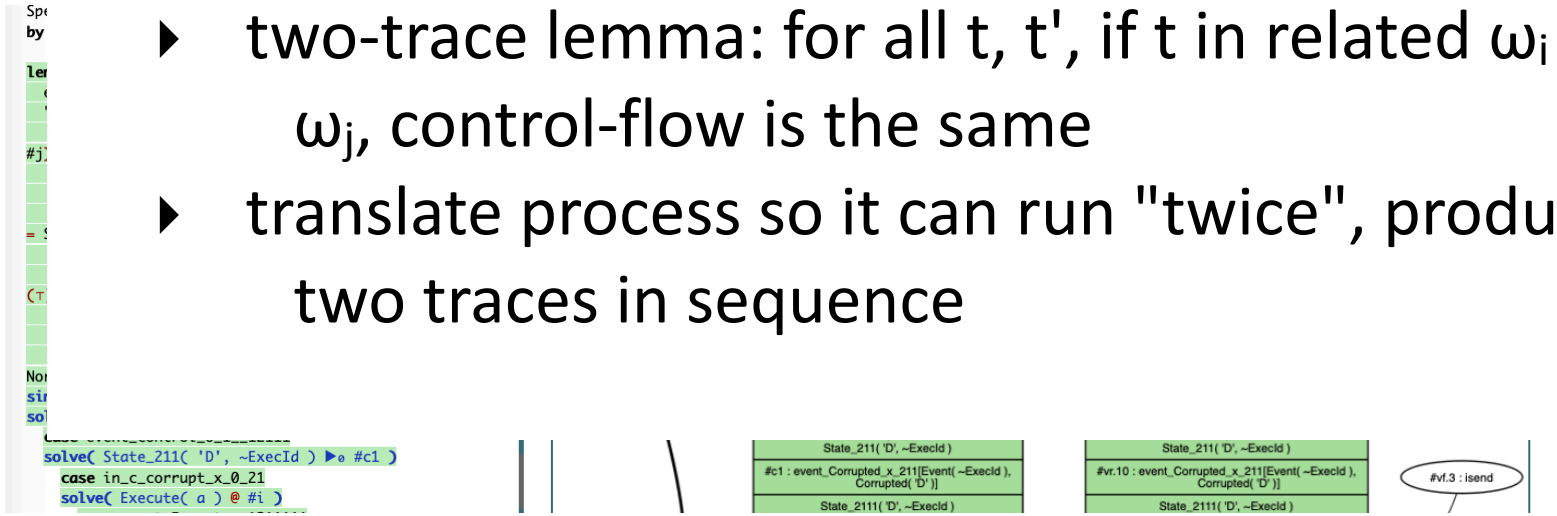
SAPIC calculus
+ verdict function
+ accountability lemmas



multiset rewrite rules
+ lemmas

attack / verification
/ timeout

- ☑ weakest possible relation
- ☑ arbitrary relation (lifting lemma offset to user)
- ☑ control-flow relation:
  - two-trace lemma: for all t, t', if t in related ω_i and ω_j, control-flow is the same
  - translate process so it can run "twice", producing two traces in sequence



Part IV case studies

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

protocol	type	# lemmas generated	# helping lemmas	time
Certificate Transp.				
model by Bruni et al	✓,r_w	31	0	41s
extended model	✓,r_w	21	0	50s
OCSP Stapling				
trusted resp.	✓,r_w	7	3	945s
untrusted resp.	✗,r_w	7	3	12s
Centralized monitor				
faulty	✗,r_c	17	0	5s
fixed	✓,r_c	17	0	3s
replication	✓,r_c	17	0	7s
Accountable alg.				
modified-1	✓,r_c	27	1	5792s
modified-2	✓,r_c	27	1	2047s

(✓): verification (✗): attack (r_w): weak relation (r_c): control-flow r.

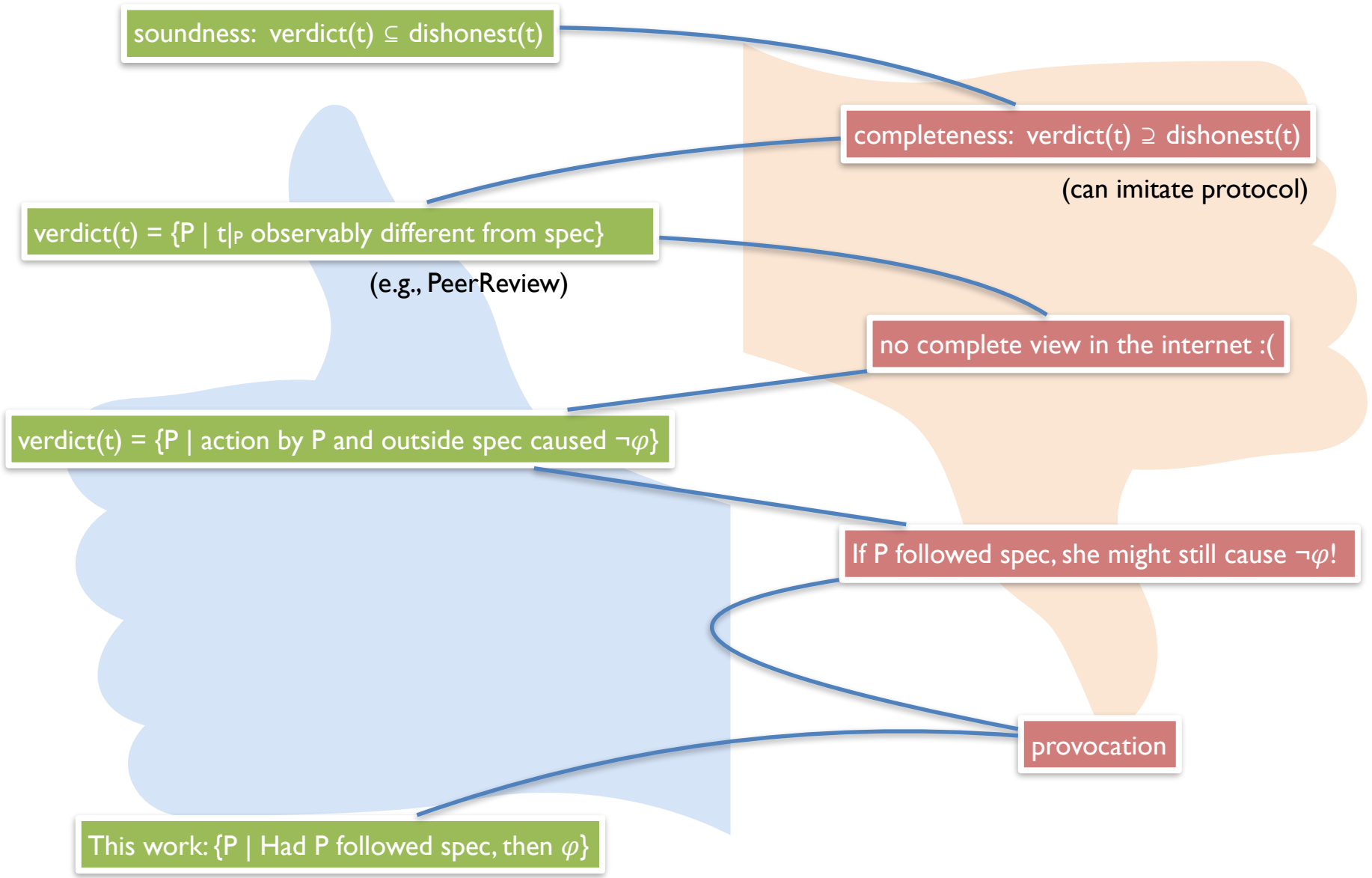
Conclusion

- Accountability is identifying misbehaving parties
- "misbehaving party" = "party whose deviation caused $\neg\varphi$"
- This definition is practical and can be verified automatically

- **Ongoing work:**
  - integrate SAPIC calculus and translation in tamarin-prover
    - see development branch
  - support arbitrary number of parties
  - accountability in the decentralised setting
    - central adversary is not w.l.o.g.!
  - accountability in the cryptographic setting
    - trace properties: 👍 indistinguishability: 🤔


Thank you!

Why is it so hard?



Case 1: weakest possible relation

- Consider t' is related to t iff corrupt(t') ⊆ corrupt(t)
- Idea: verdict function defined as

$$verdict(t) = \begin{cases} V_1 & if\ \omega_1(t) \\ \vdots & \\ V_n & if\ \omega_n(t) \end{cases}$$

- cases are **exhaustive** and **exclusive**
- **sufficiency**: $S \in V_i \Rightarrow \exists t.$ corrupted(t)=S and $\neg\varphi(t)$
- **verifiability**: $V_i = \varnothing \iff \varphi$
- **minimality**: can't do with less than $S \in V_i$
- **uniqueness**: whenever ω_i is observed, parties in V_i are corrupted
- **completeness**: (.. left out ..)

Conclusion

- Accountability via causation works and can be verified automatically

- **Ongoing work:**
  - integrate SAPIC calculus and translation in tamarin-prover
  - support arbitrary number of parties
- **Accountability in the decentralised setting (unpublished work)**
  - original definition in decentralised setting, parties deviate individually
  - provocation problem → centralised setting is not w.l.o.g.!
  - optimality requirement: deviating parties exchange no more information than necessary. conjectured to be equal to centralised setting.