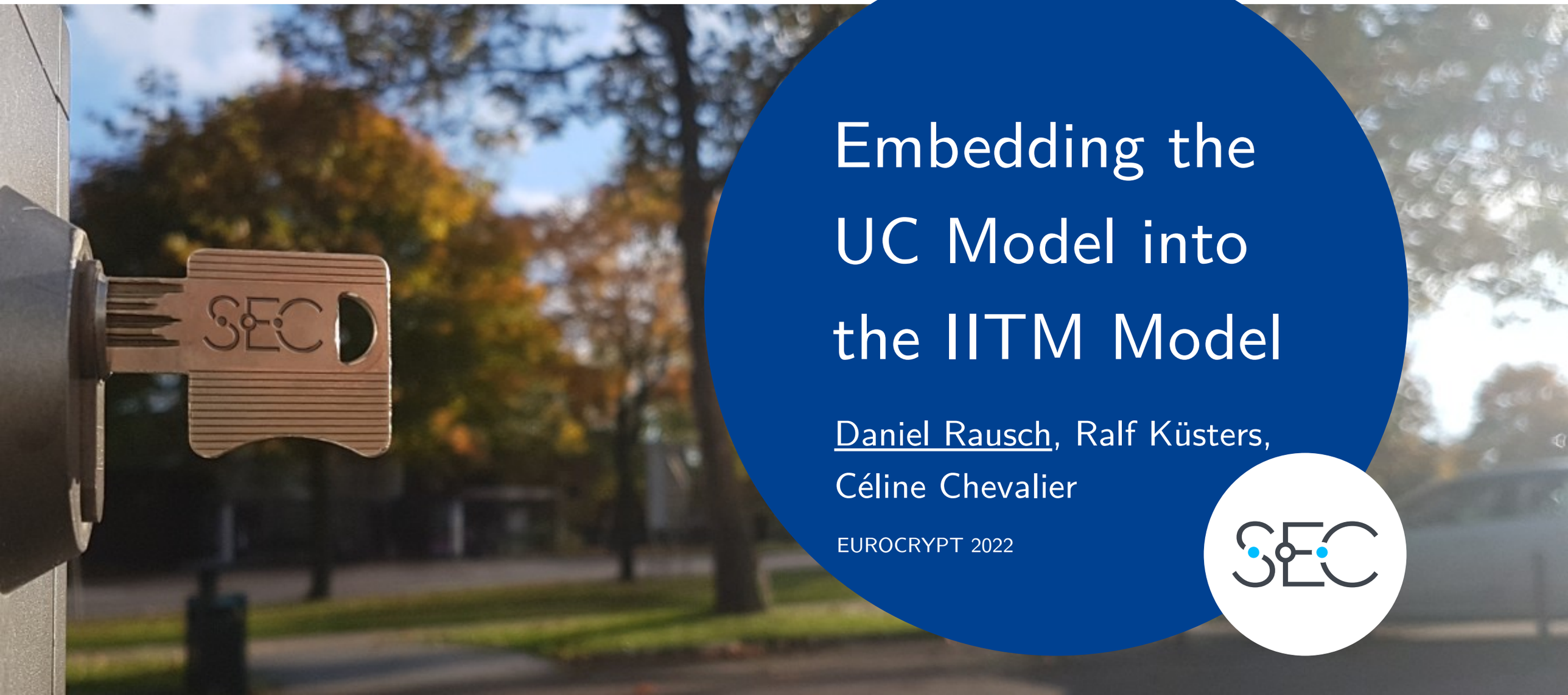
University of Stuttgart
Institute of
Information Security



Embedding the UC Model into the IITM Model

Daniel Rausch, Ralf Küsters,
Céline Chevalier

EUROCRYPT 2022

Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

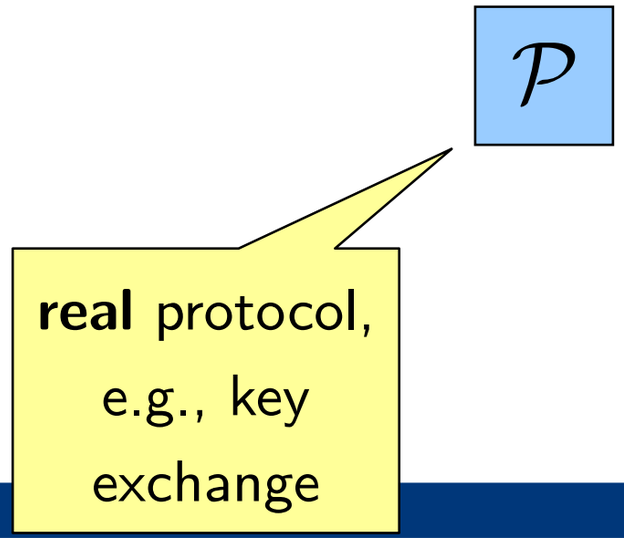
UC security:

Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

UC security:

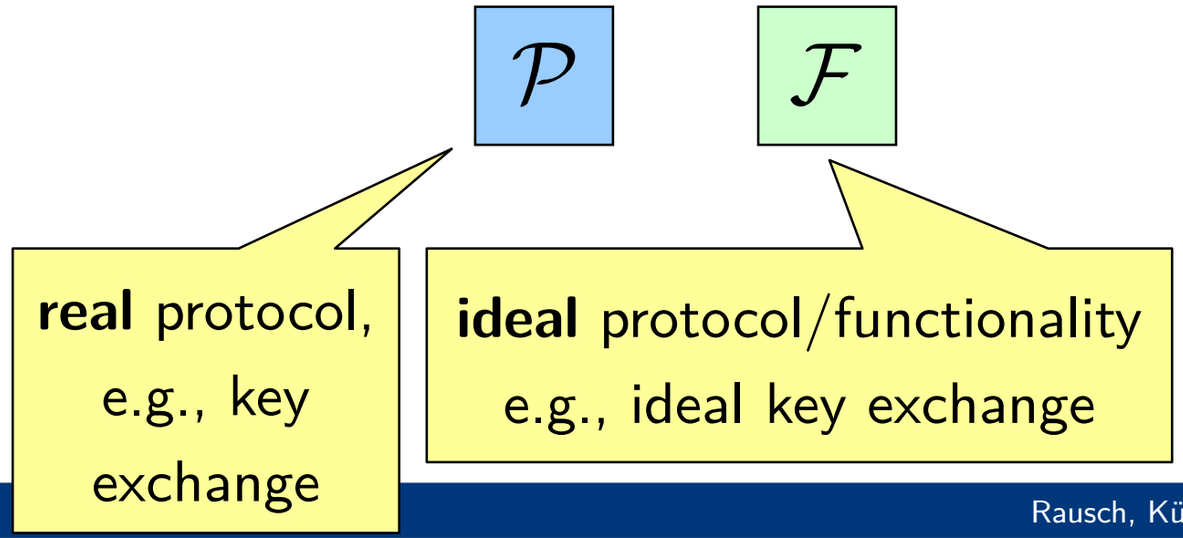


Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

UC security:

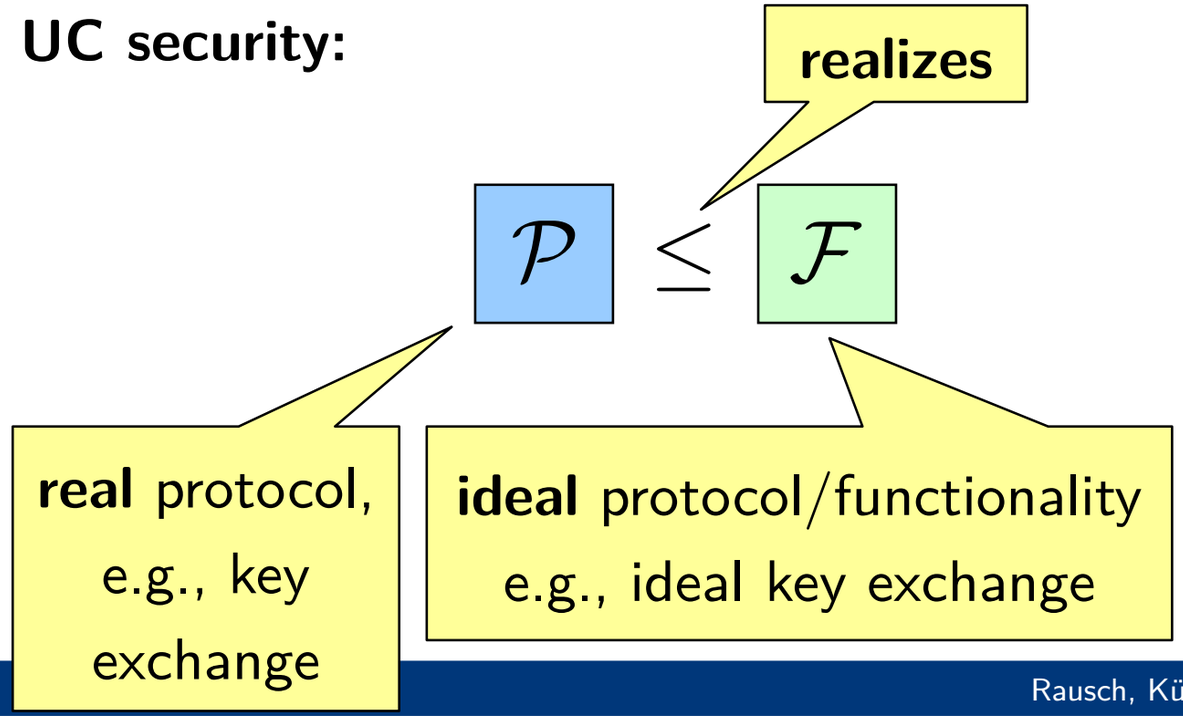


Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

UC security:

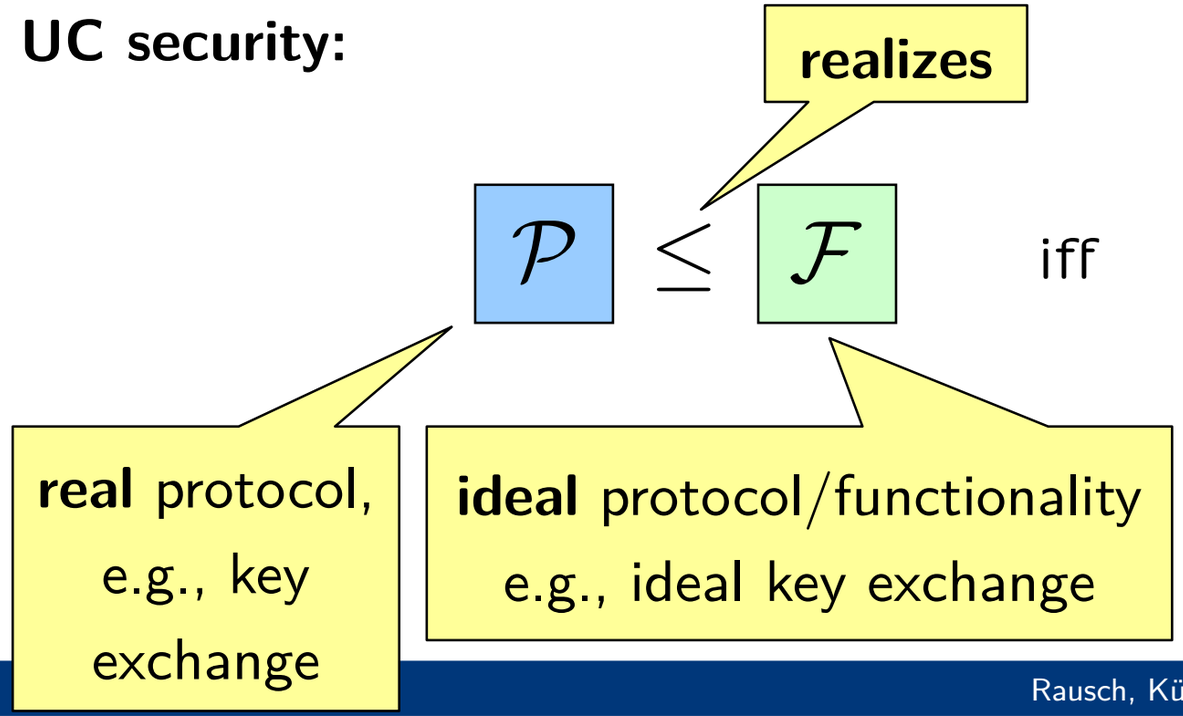


Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

UC security:

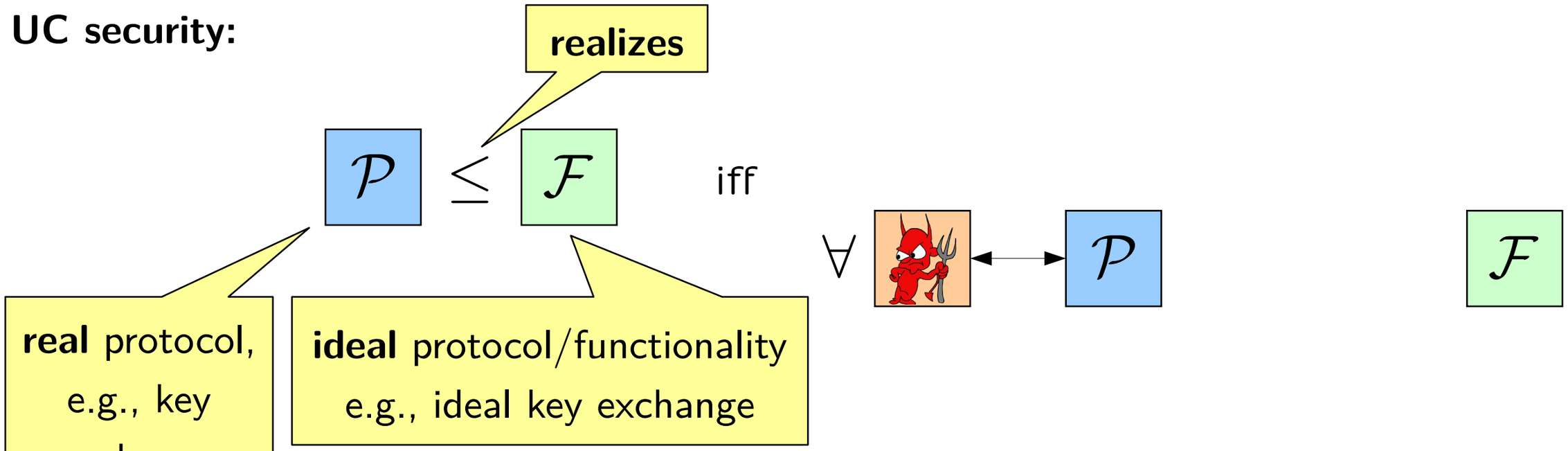


Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

UC security:

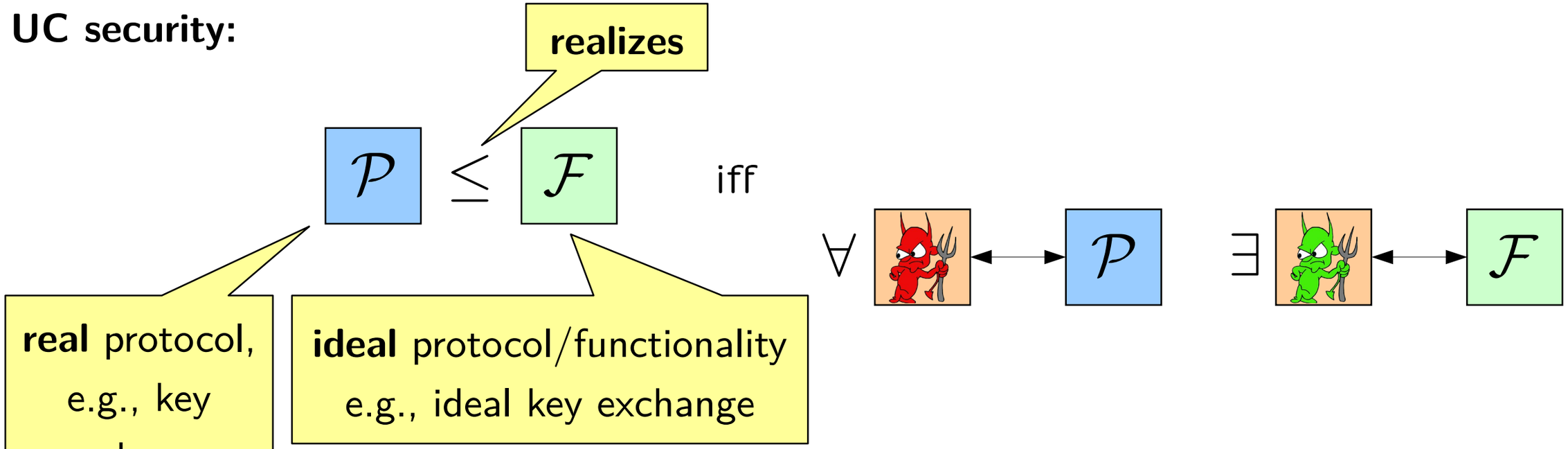


Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

UC security:

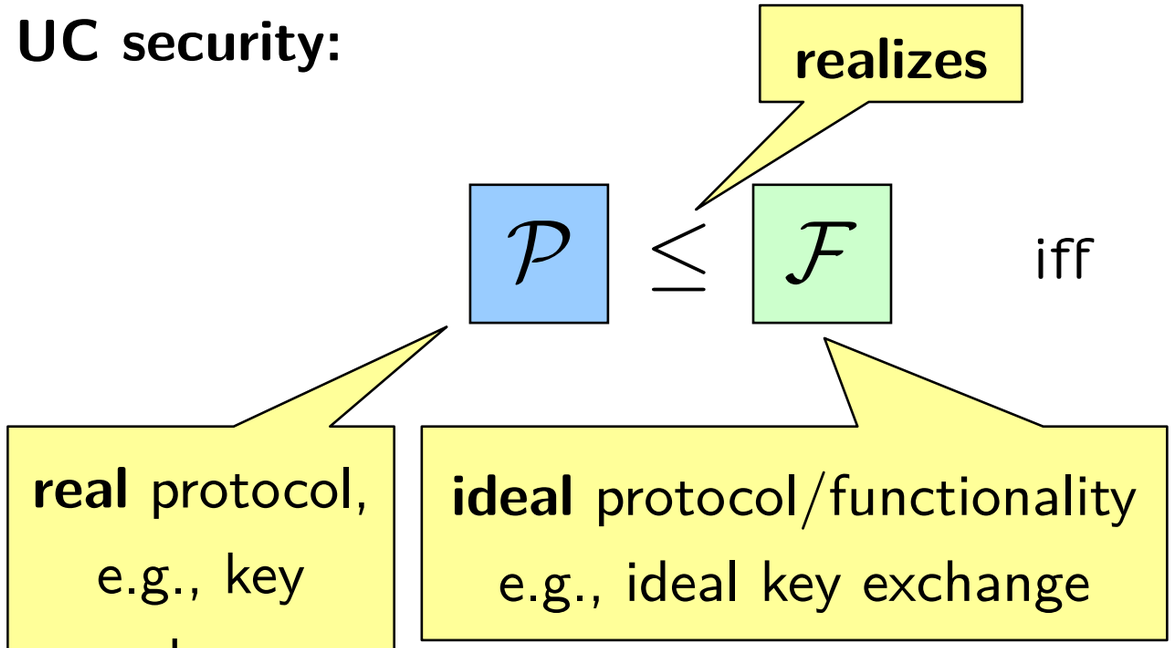


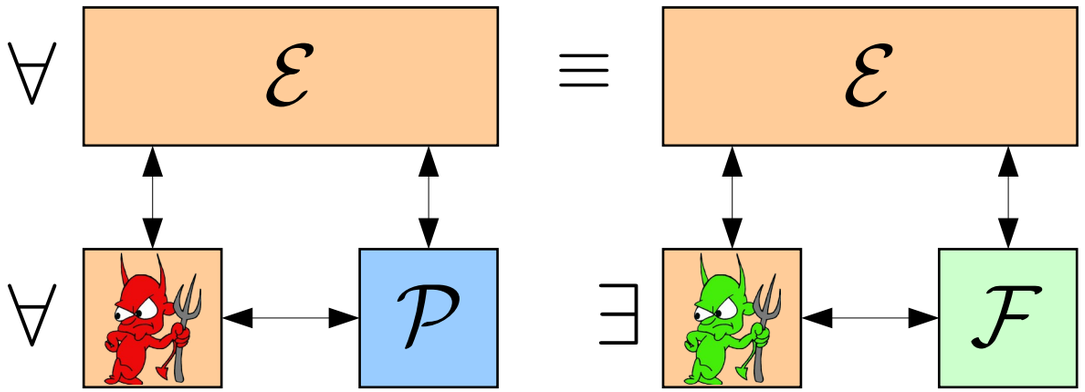
Universal Composability

Widely used concept for defining and analyzing protocol security:

- Strong security guarantees
- Composability via composition theorems
  - Modular analysis
  - Re-use results

UC security:



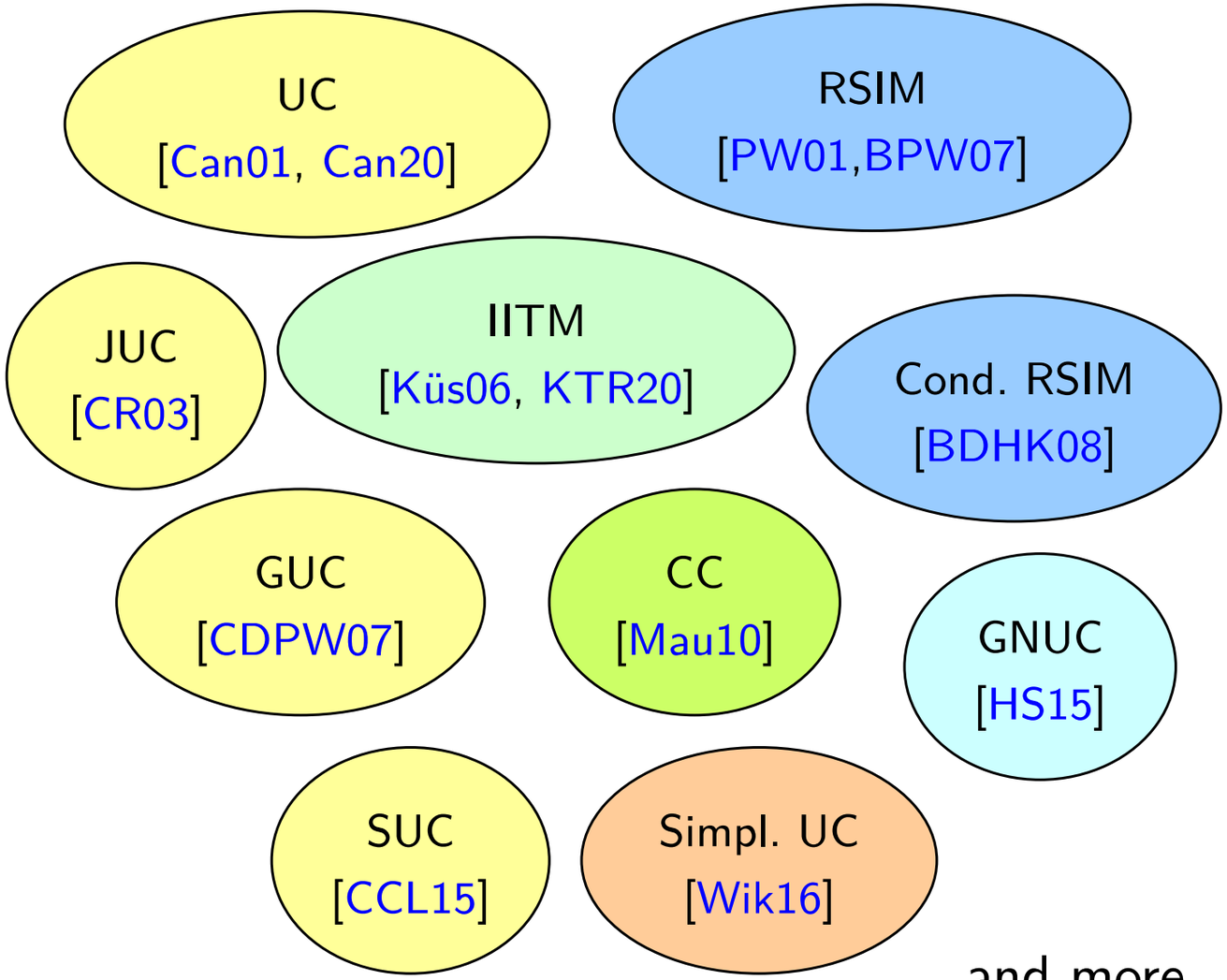


Models for Universal Composability

Numerous models:

Models for Universal Composability

Numerous models:

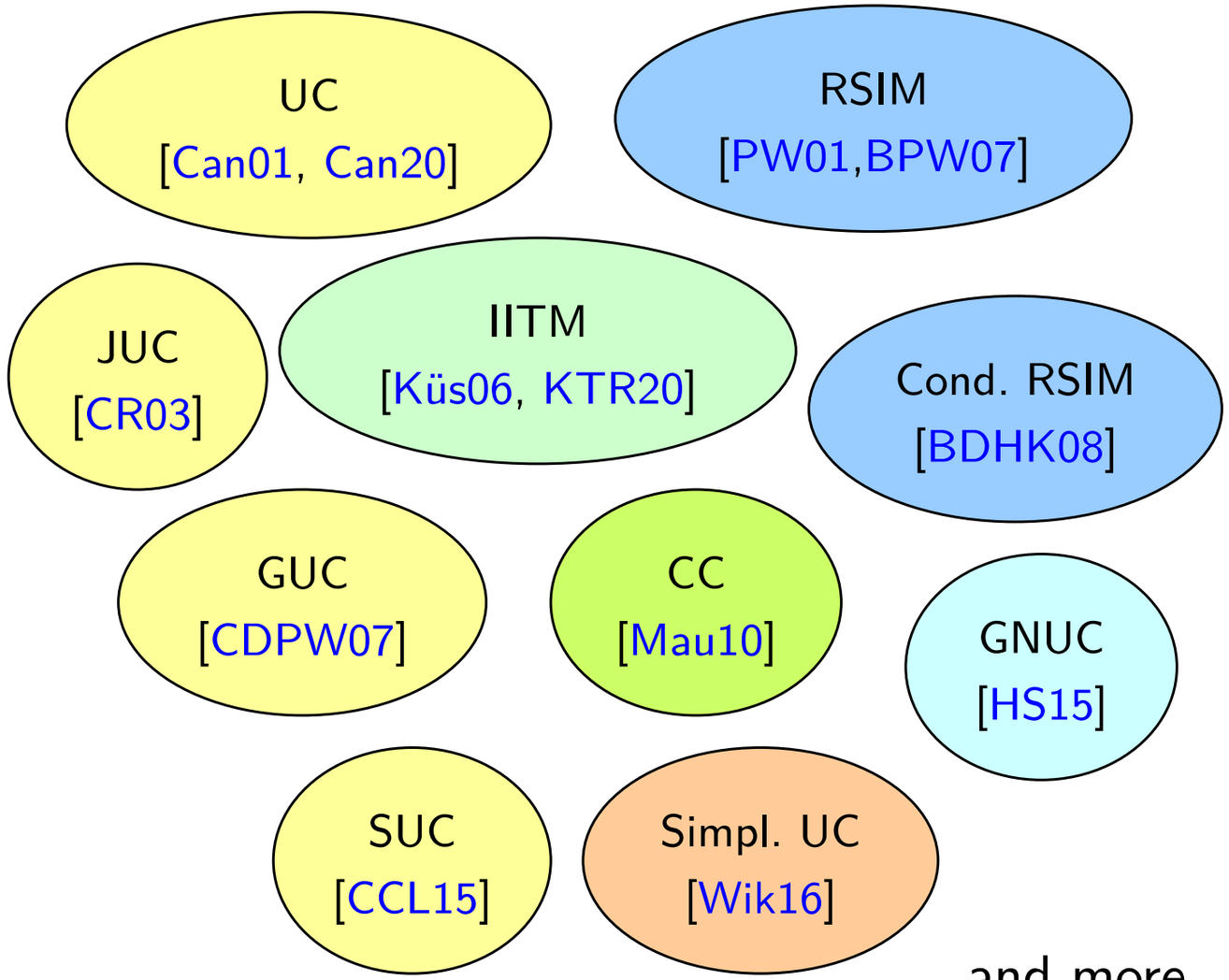


and more...

Models for Universal Composability

Numerous models:

- Same basic idea but differ drastically in technical implementation and features

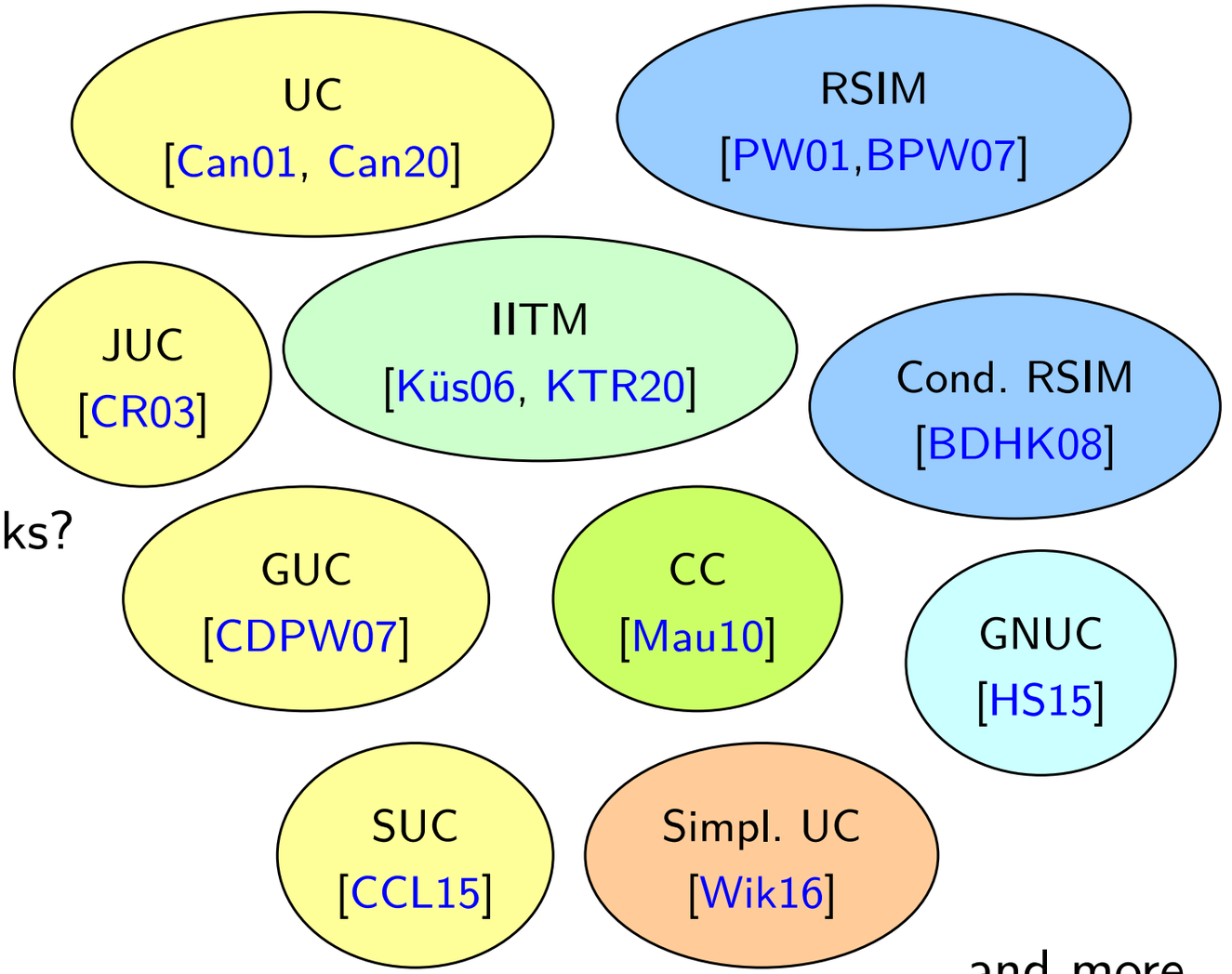


and more...

Models for Universal Composability

Numerous models:

- Same basic idea but differ drastically in technical implementation and features
- Relationships so far unexplored:
  - Expressiveness?
  - Strength of security results? Missed attacks?
  - Re-usability of results?



and more...

Models for Universal Composability

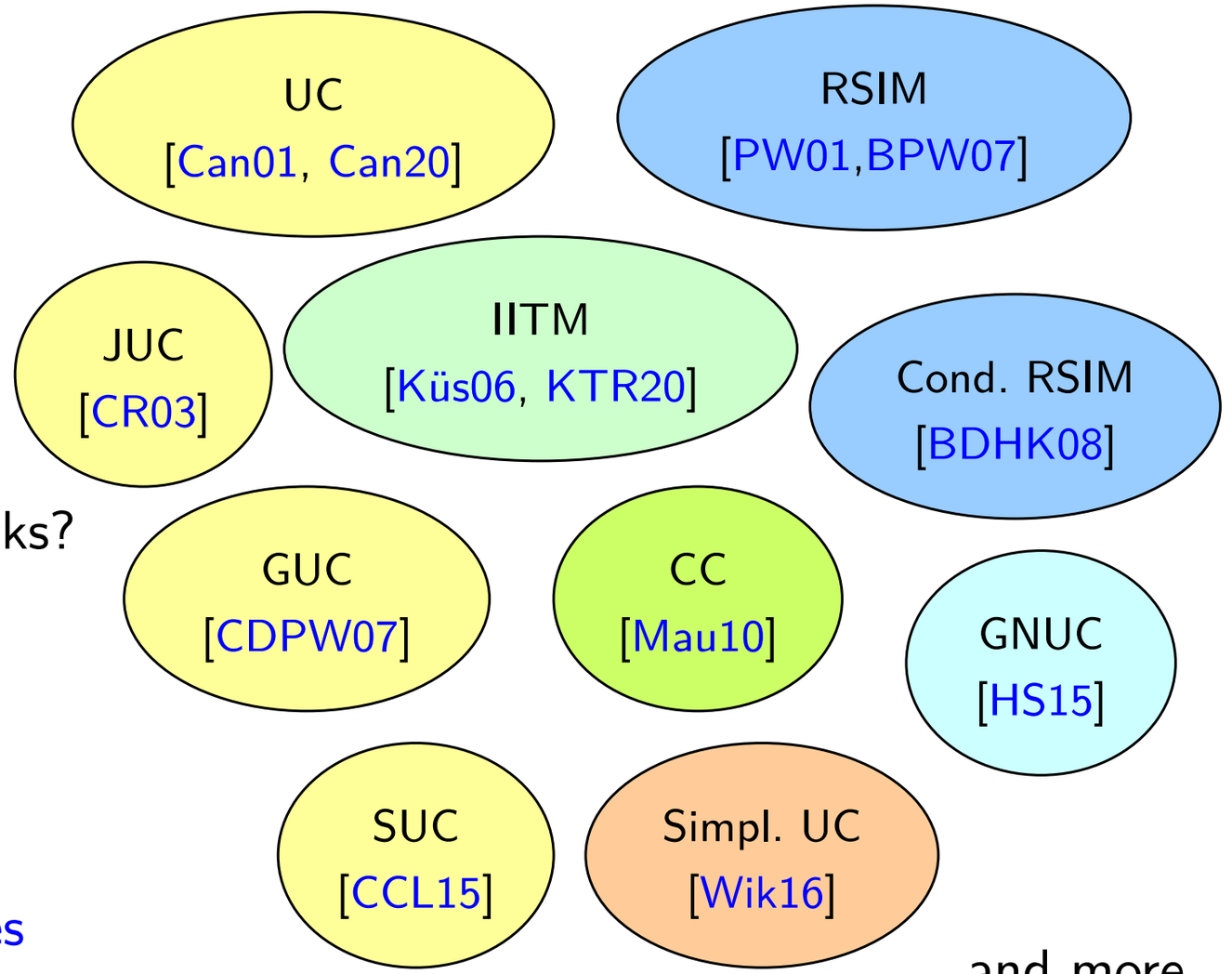
Numerous models:

- Same basic idea but differ drastically in technical implementation and features
- Relationships so far unexplored:
  - Expressiveness?
  - Strength of security results? Missed attacks?
  - Re-usability of results?

Our goal: Formally relate models

→ Educated choice of model

→ Map and re-use protocols, results, features from one model to others



Models for Universal Composability

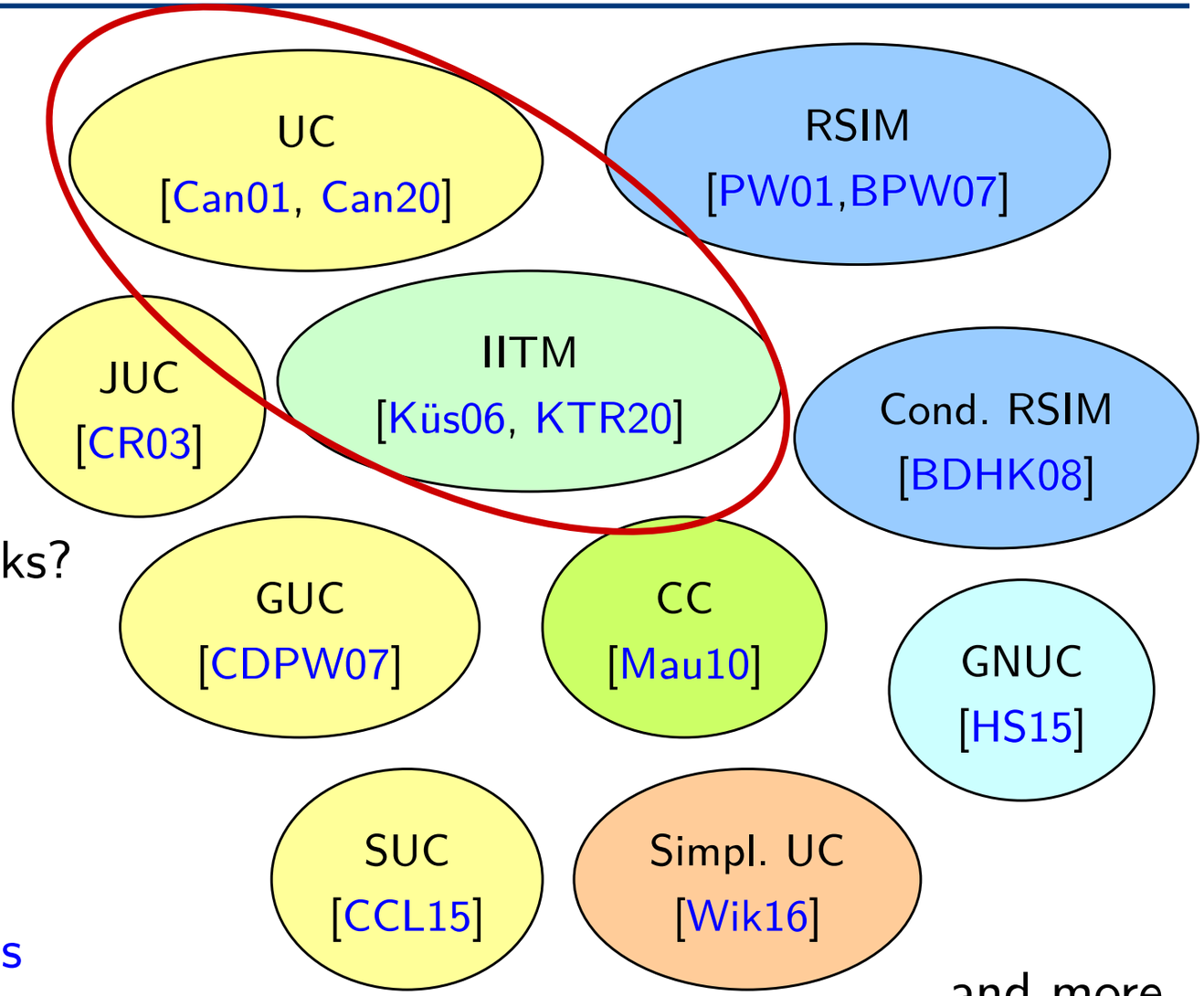
Numerous models:

- Same basic idea but differ drastically in technical implementation and features
- Relationships so far unexplored:
  - Expressiveness?
  - Strength of security results? Missed attacks?
  - Re-usability of results?

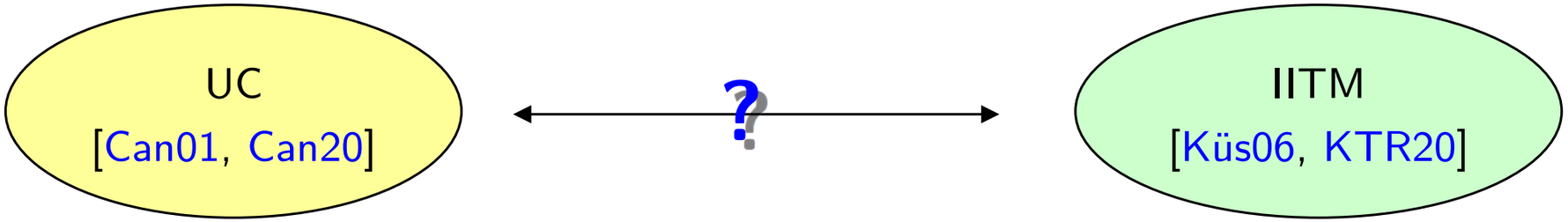
Our goal: Formally relate models

→ Educated choice of model

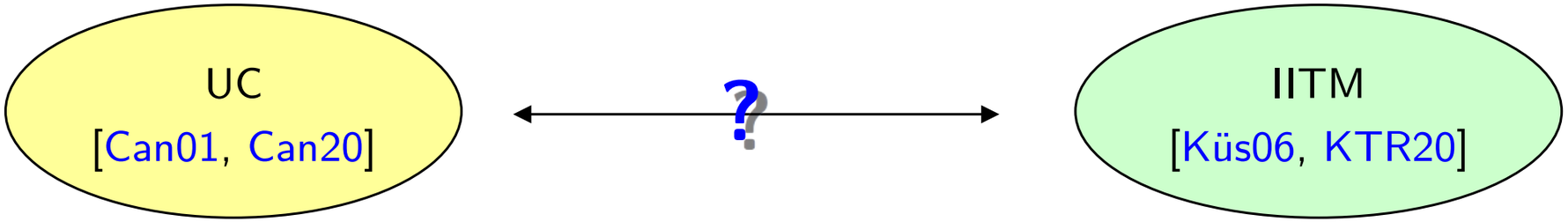
→ Map and re-use protocols, results, features from one model to others



Why UC and IITM?

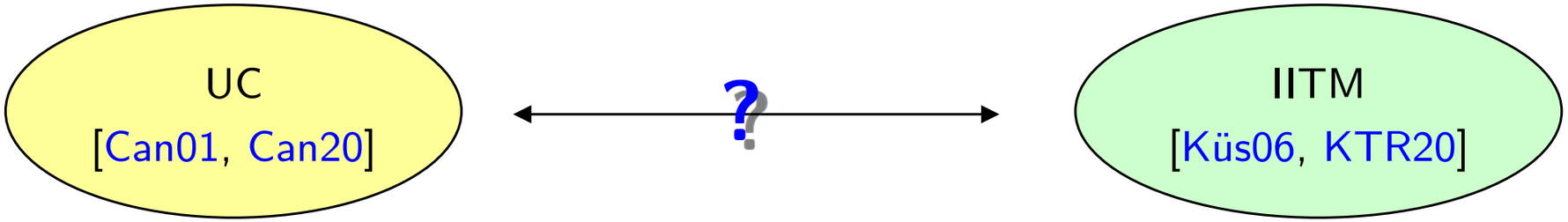


Why UC and IITM?



- Most commonly used
- Extensive literature: protocols and security results in a wide range of settings

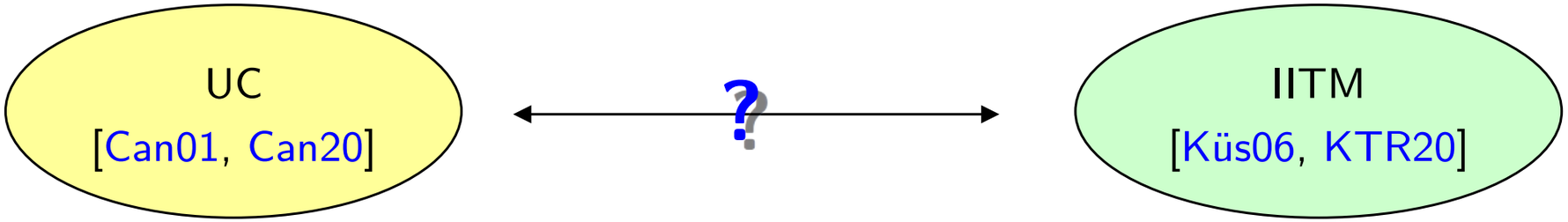
Why UC and IITM?



- Most commonly used
- Extensive literature: protocols and security results in a wide range of settings

- Many interesting features, e.g., seamless support for protocols/composition with:
  - joint, global, arbitrarily shared state
  - globally shared SIDs, locally managed SIDs
  - combinations of all of the above

Why UC and IITM?



- Most commonly used
- Extensive literature: protocols and security results in a wide range of settings

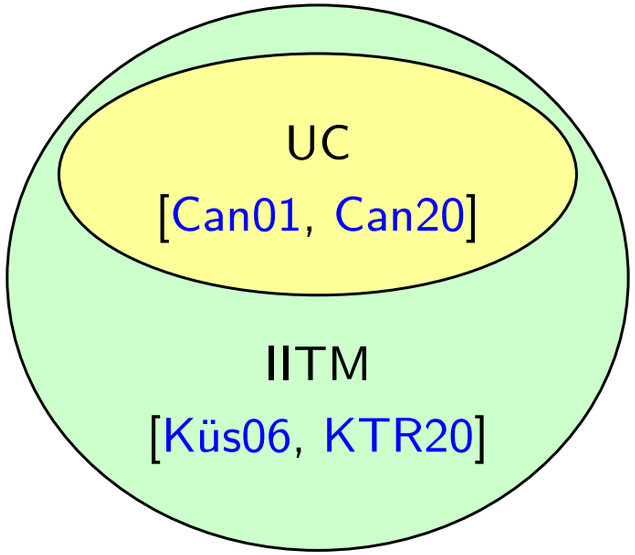
- Many interesting features, e.g., seamless support for protocols/composition with:
  - joint, global, arbitrarily shared state
  - globally shared SIDs, locally managed SIDs
  - combinations of all of the above
- Has also been applied to protocols not yet captured in UC

Overview of Our Contributions

Main contributions:

Overview of Our Contributions

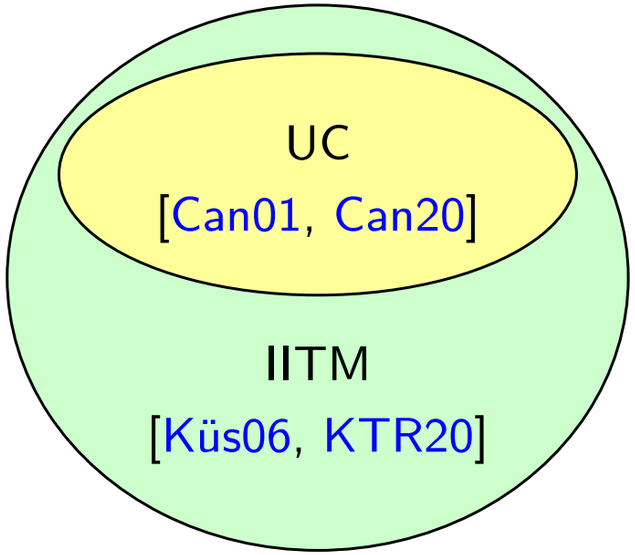
Main contributions:



Overview of Our Contributions

Main contributions:

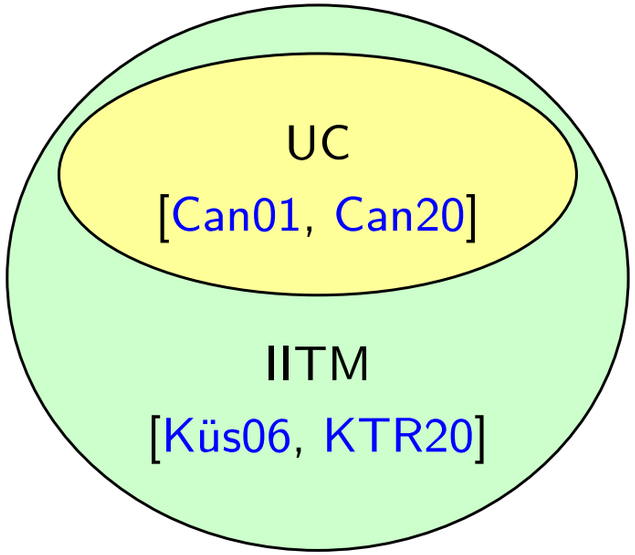
- Relate different concepts



Overview of Our Contributions

Main contributions:

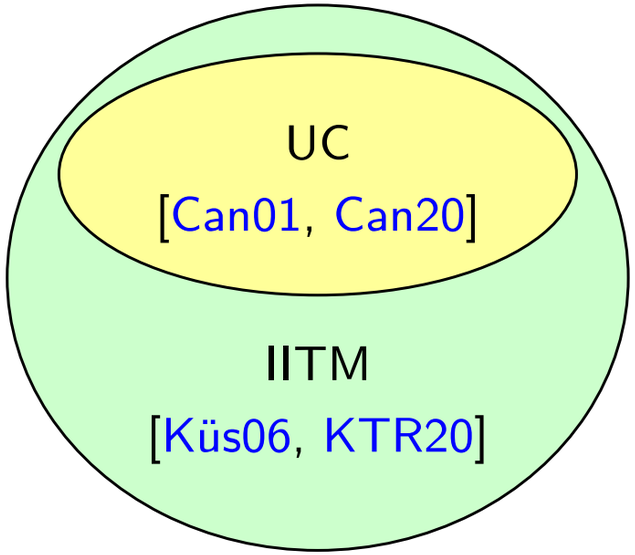
- Relate different concepts
- Mapping from arbitrary UC protocols to IITM protocols



Overview of Our Contributions

Main contributions:

- Relate different concepts
- Mapping from arbitrary UC protocols to IITM protocols
- Mapping preserves security and composability results

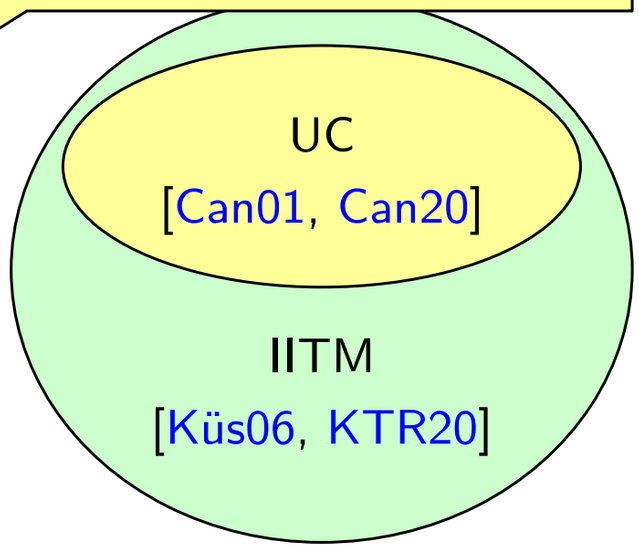


Overview of Our Contributions

Main contributions:

- Relate different concepts
- Mapping from arbitrary UC protocols to IITM protocols
- Mapping preserves security and composability results

Immediate practical benefit: Combine existing UC results with IITM features

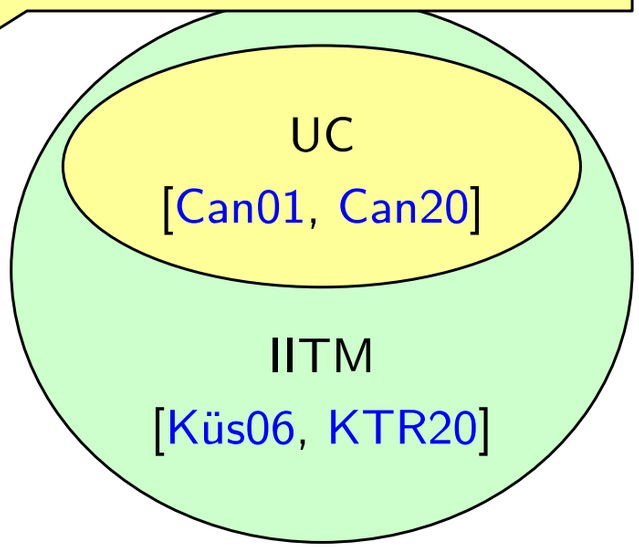


Overview of Our Contributions

Main contributions:

- Relate different concepts
- Mapping from arbitrary UC protocols to IITM protocols
- Mapping preserves security and composability results
- Other direction is impossible in general

Immediate practical benefit: Combine existing UC results with IITM features

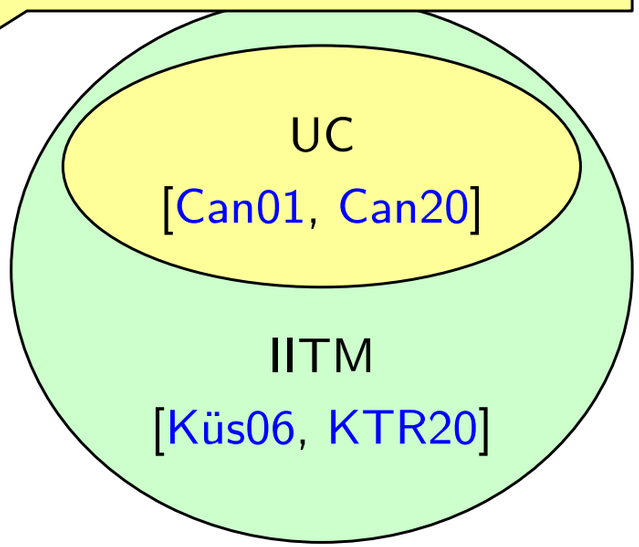


Overview of Our Contributions

Main contributions:

- Relate different concepts
- Mapping from arbitrary UC protocols to IITM protocols
- Mapping preserves security and composability results
- Other direction is impossible in general

Immediate practical benefit: Combine existing UC results with IITM features



Further results (details in the paper):

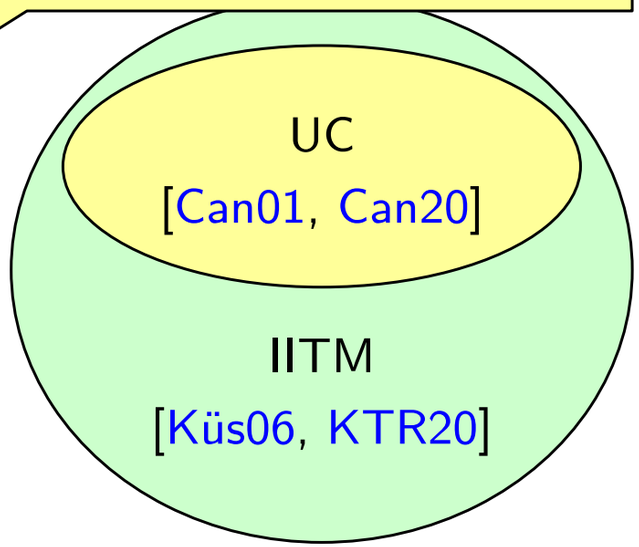
- Find and fix several issues in the UC model that formally prevent composition

Overview of Our Contributions

Main contributions:

- Relate different concepts
- Mapping from arbitrary UC protocols to IITM protocols
- Mapping preserves security and composability results
- Other direction is impossible in general

Immediate practical benefit: Combine existing UC results with IITM features



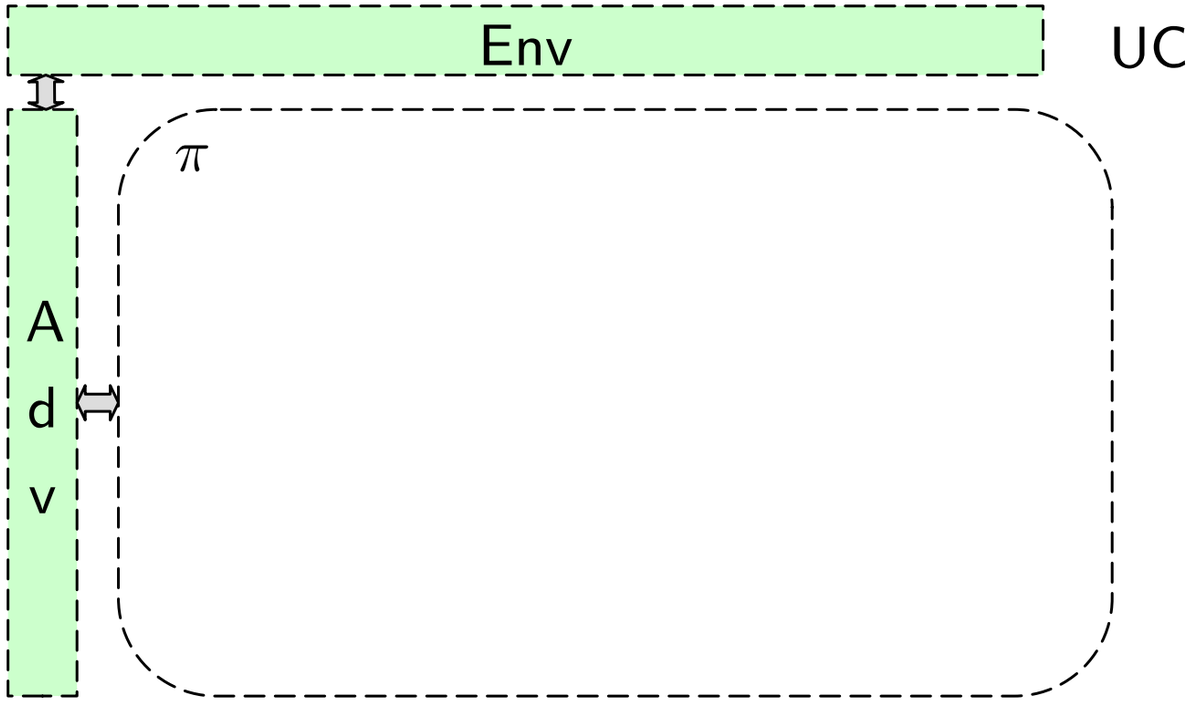
Further results (details in the paper):

- Find and fix several issues in the UC model that formally prevent composition
- Modeling technique for a new type of composition

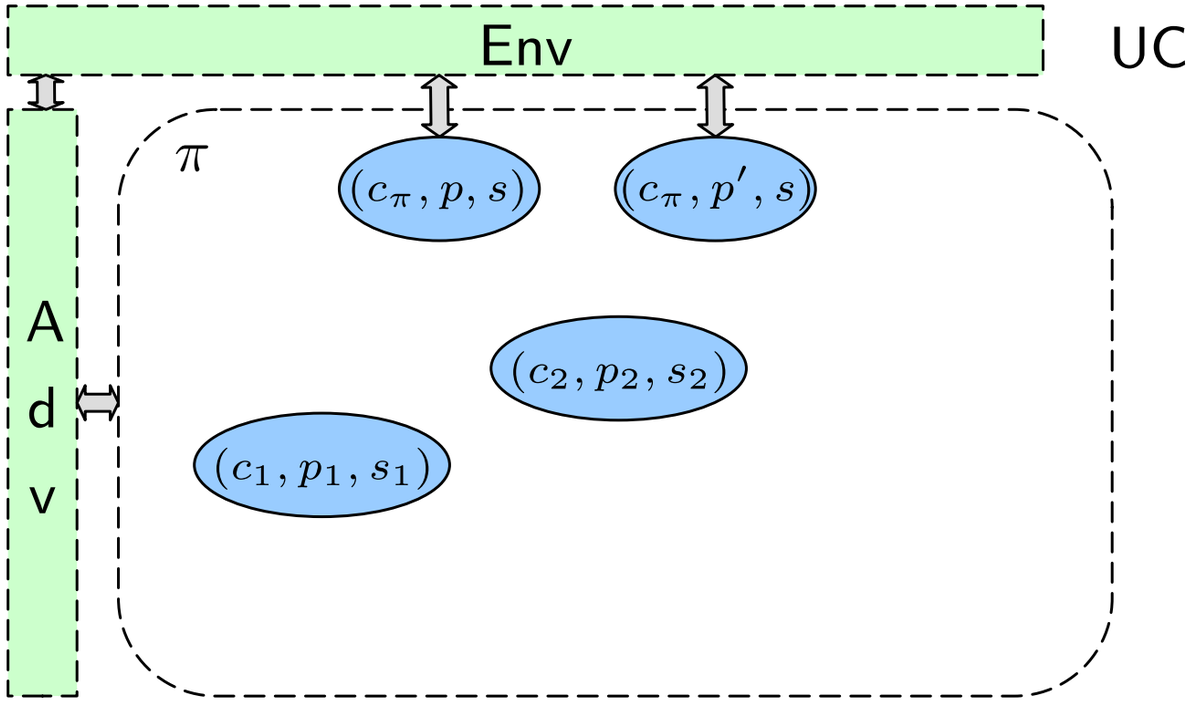
Main Contributions

- **Concepts of UC and IITM**
- Embedding UC into IITM
- The Other Direction

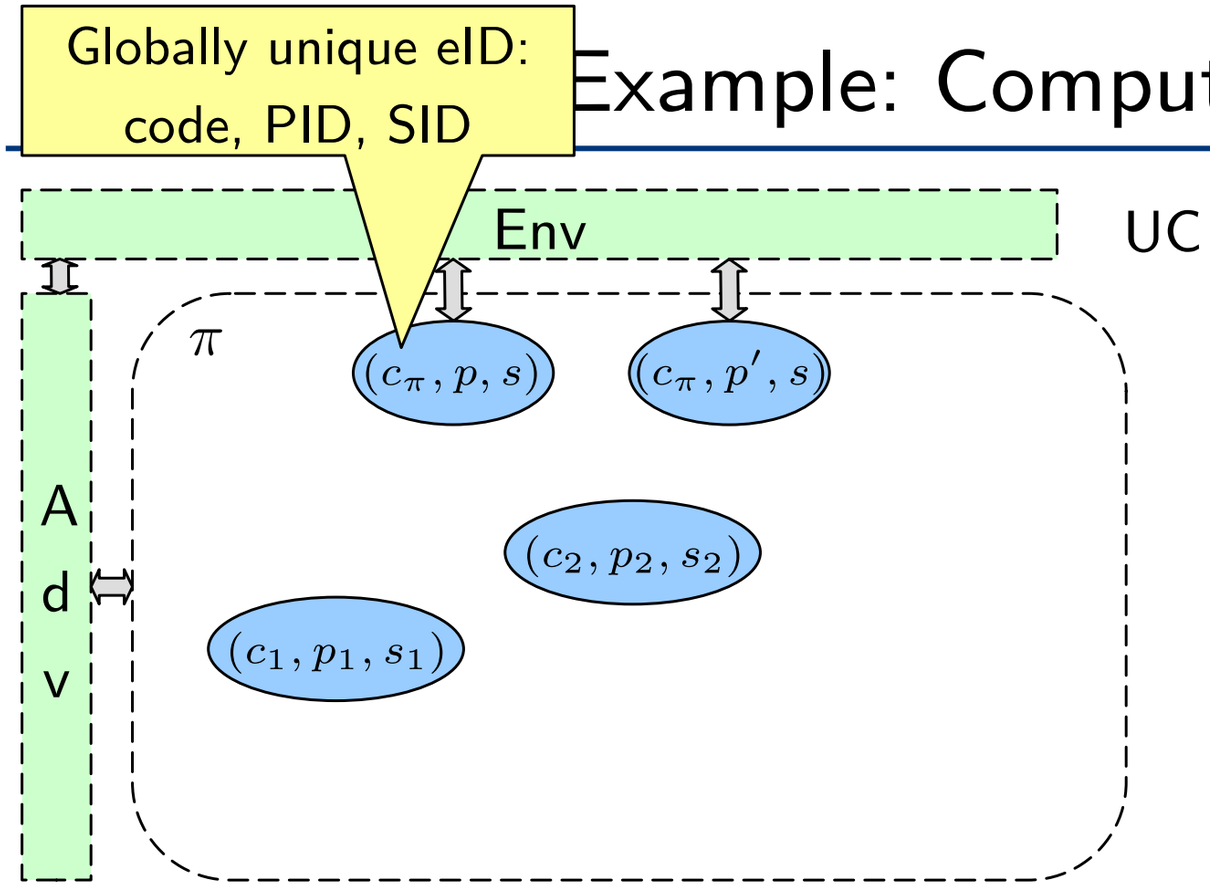
Example: Computational Framework



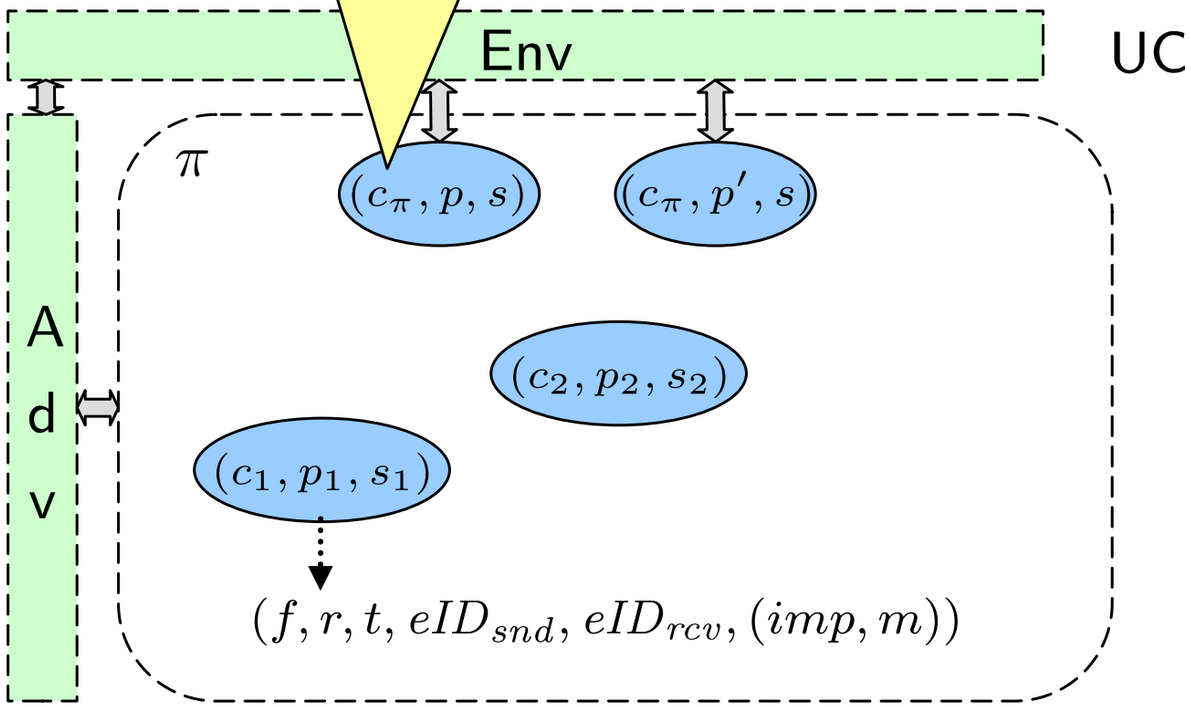
Example: Computational Framework



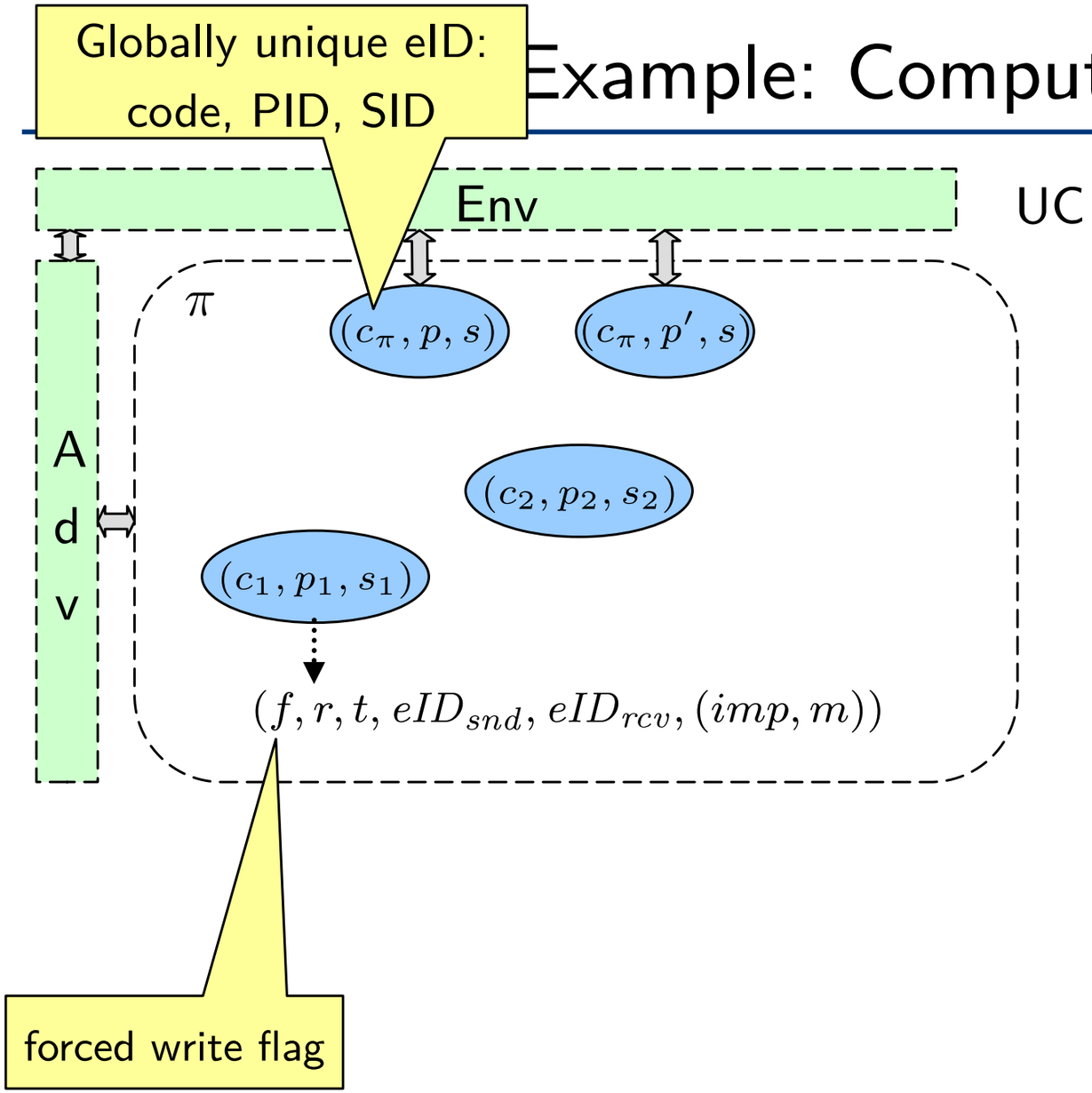
Example: Computational Framework



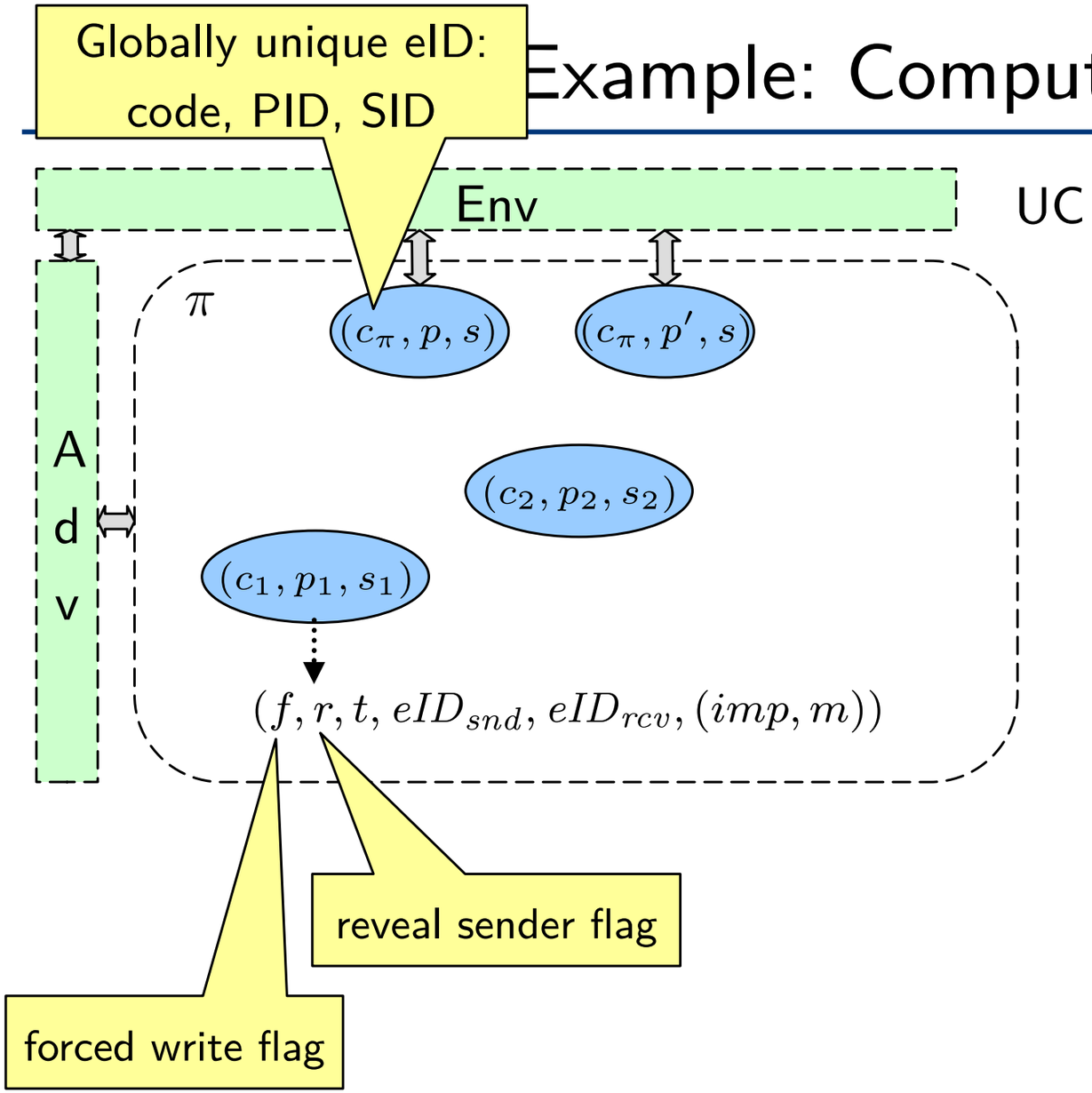
Example: Computational Framework



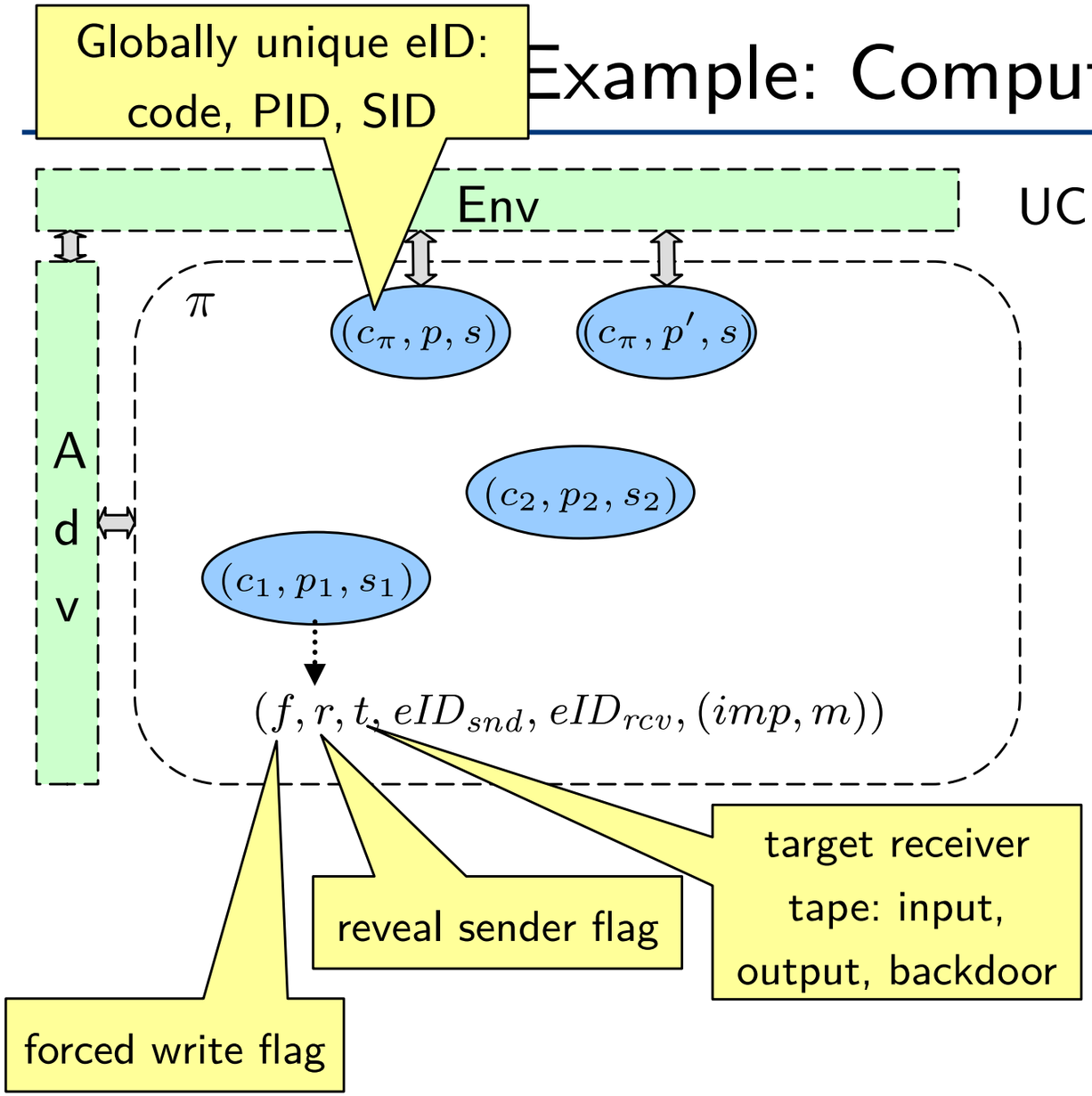
Example: Computational Framework



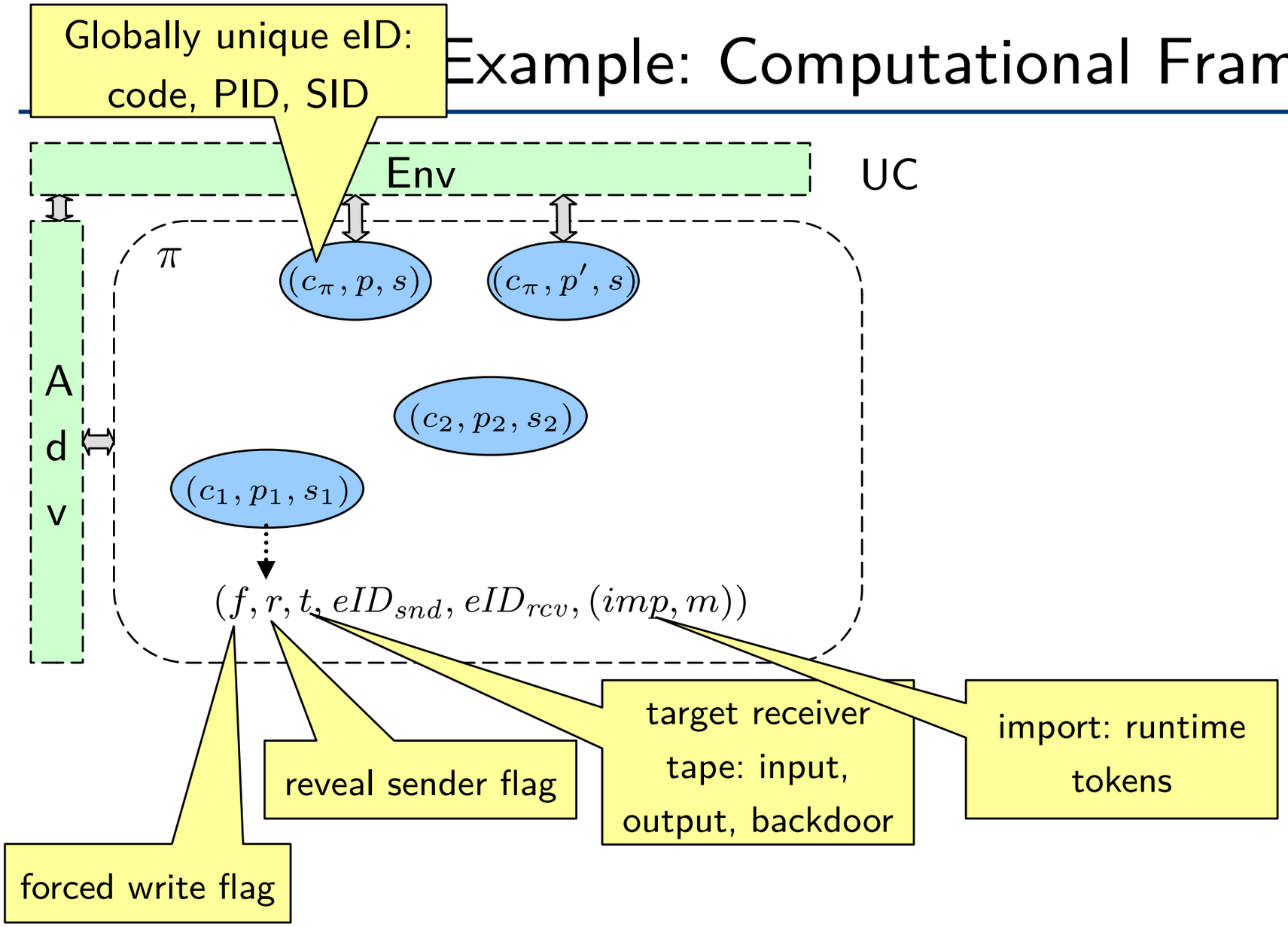
Example: Computational Framework



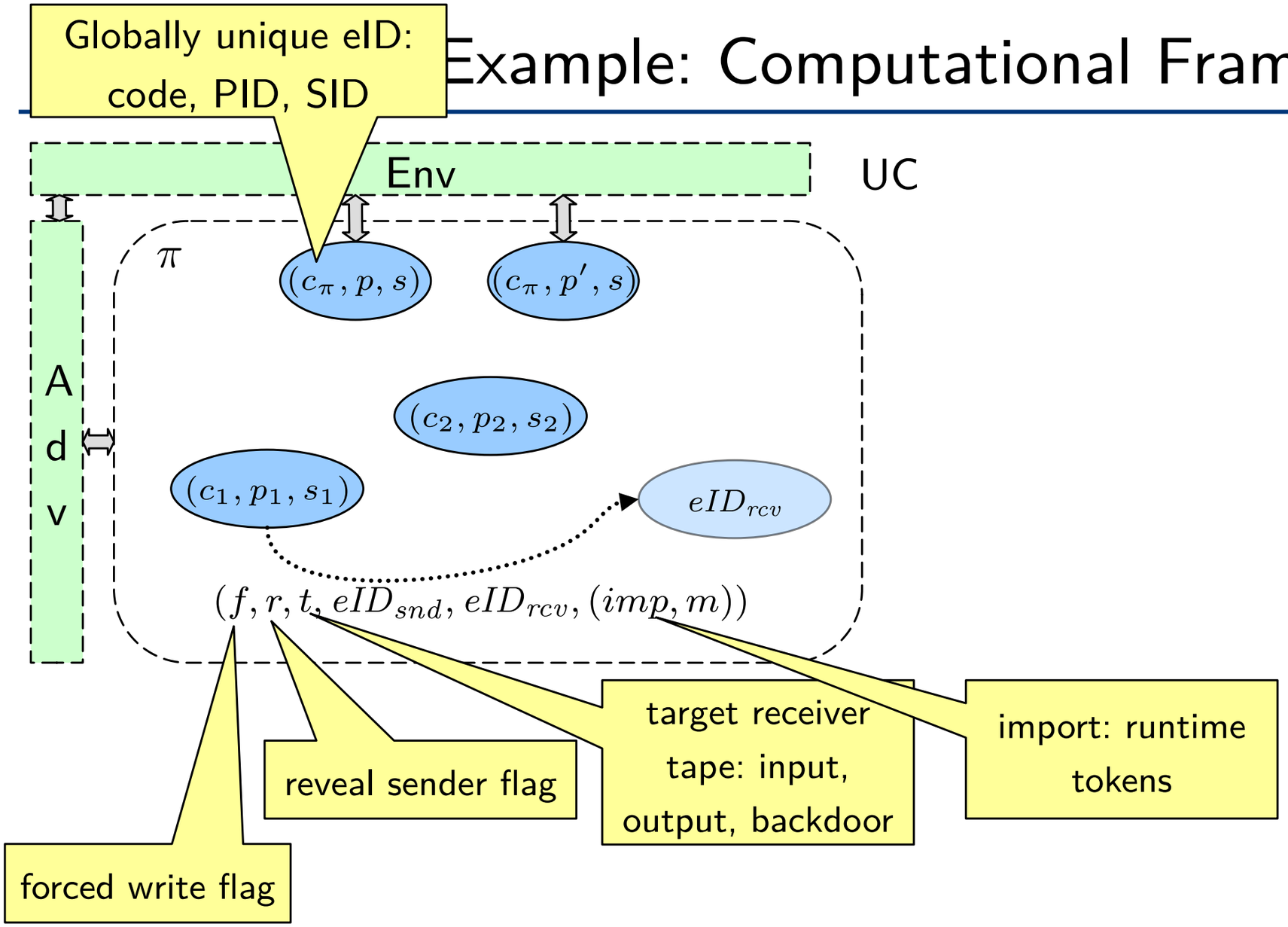
Example: Computational Framework



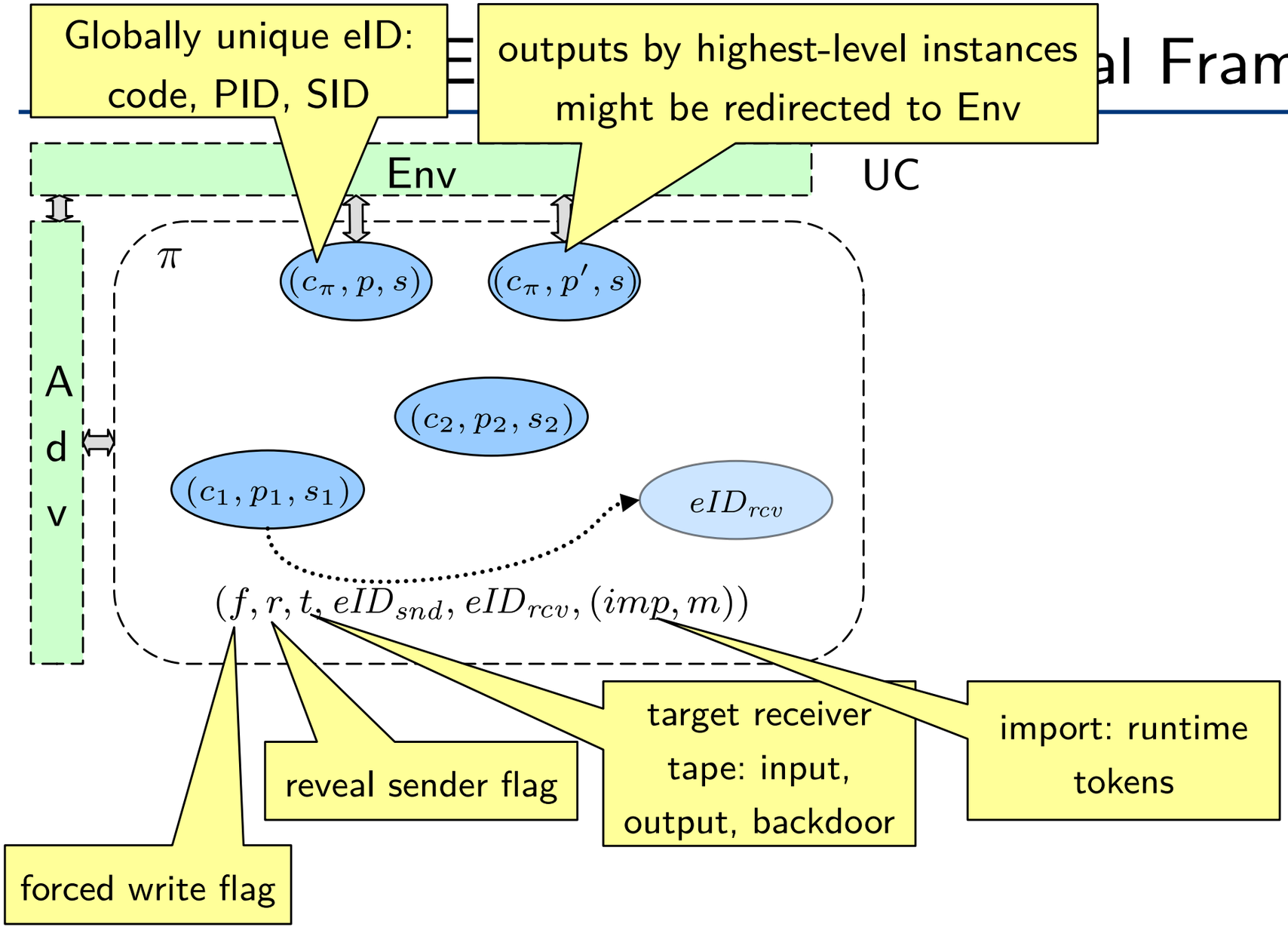
Example: Computational Framework



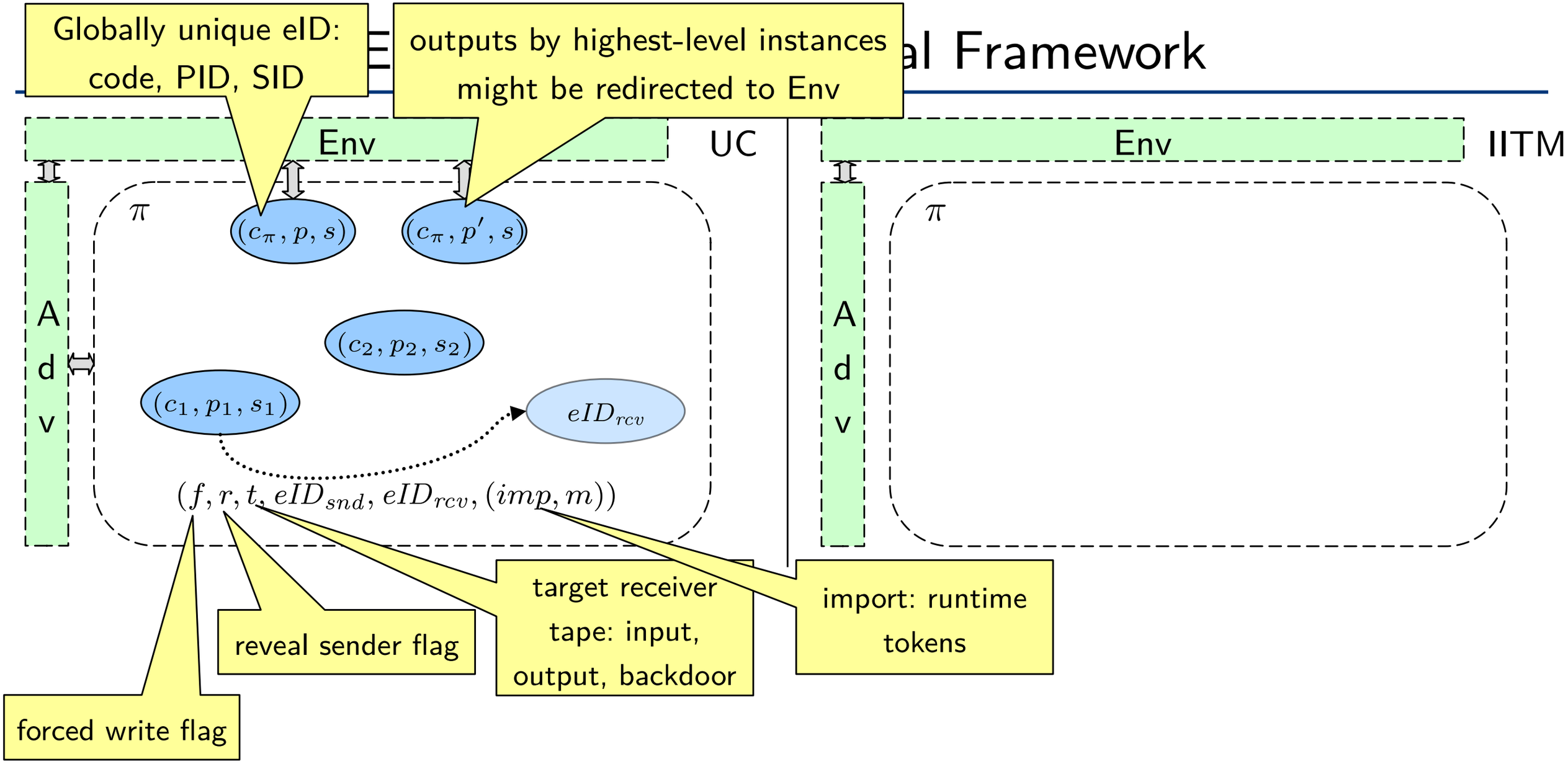
Example: Computational Framework



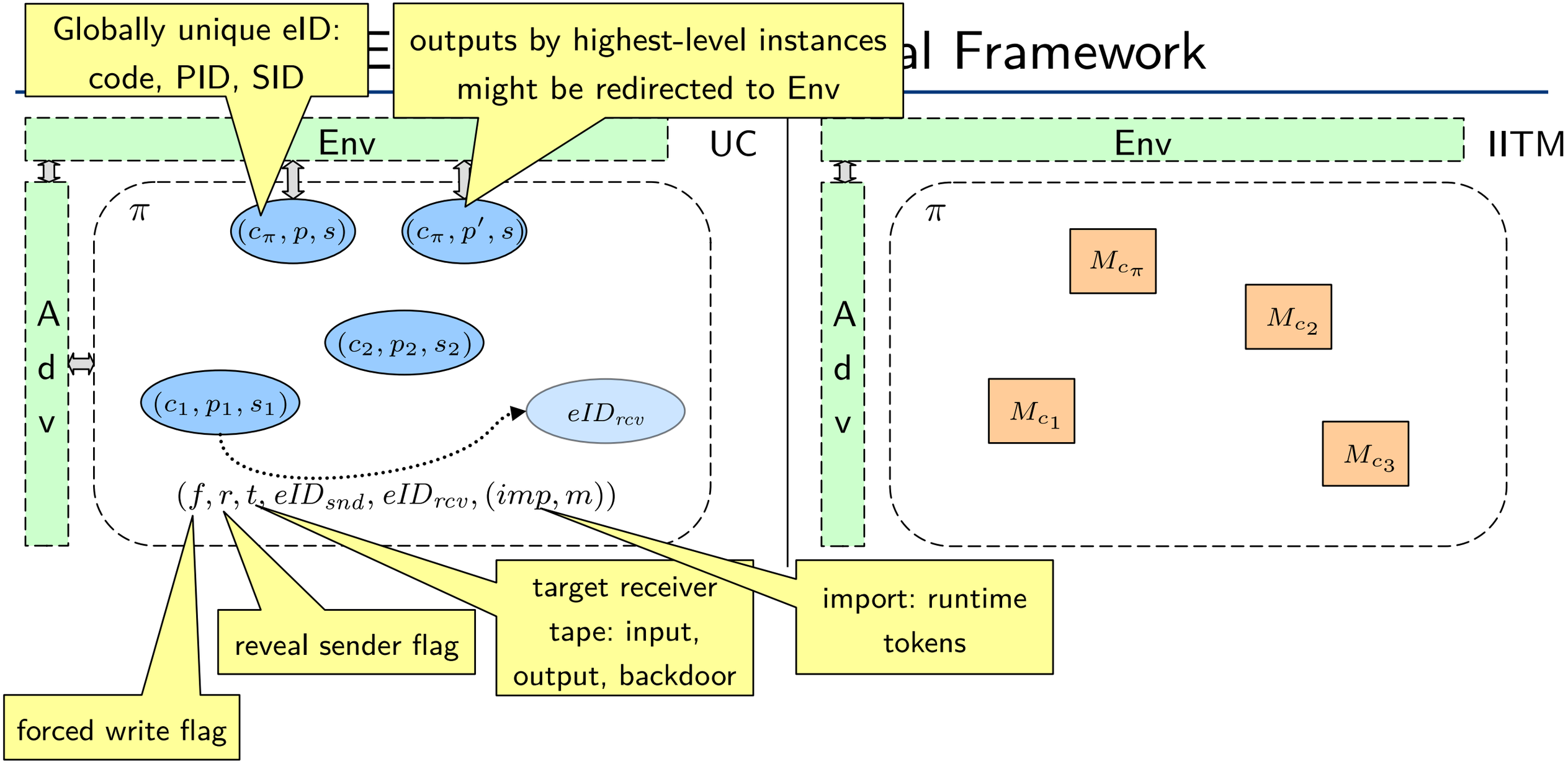
E al Framework



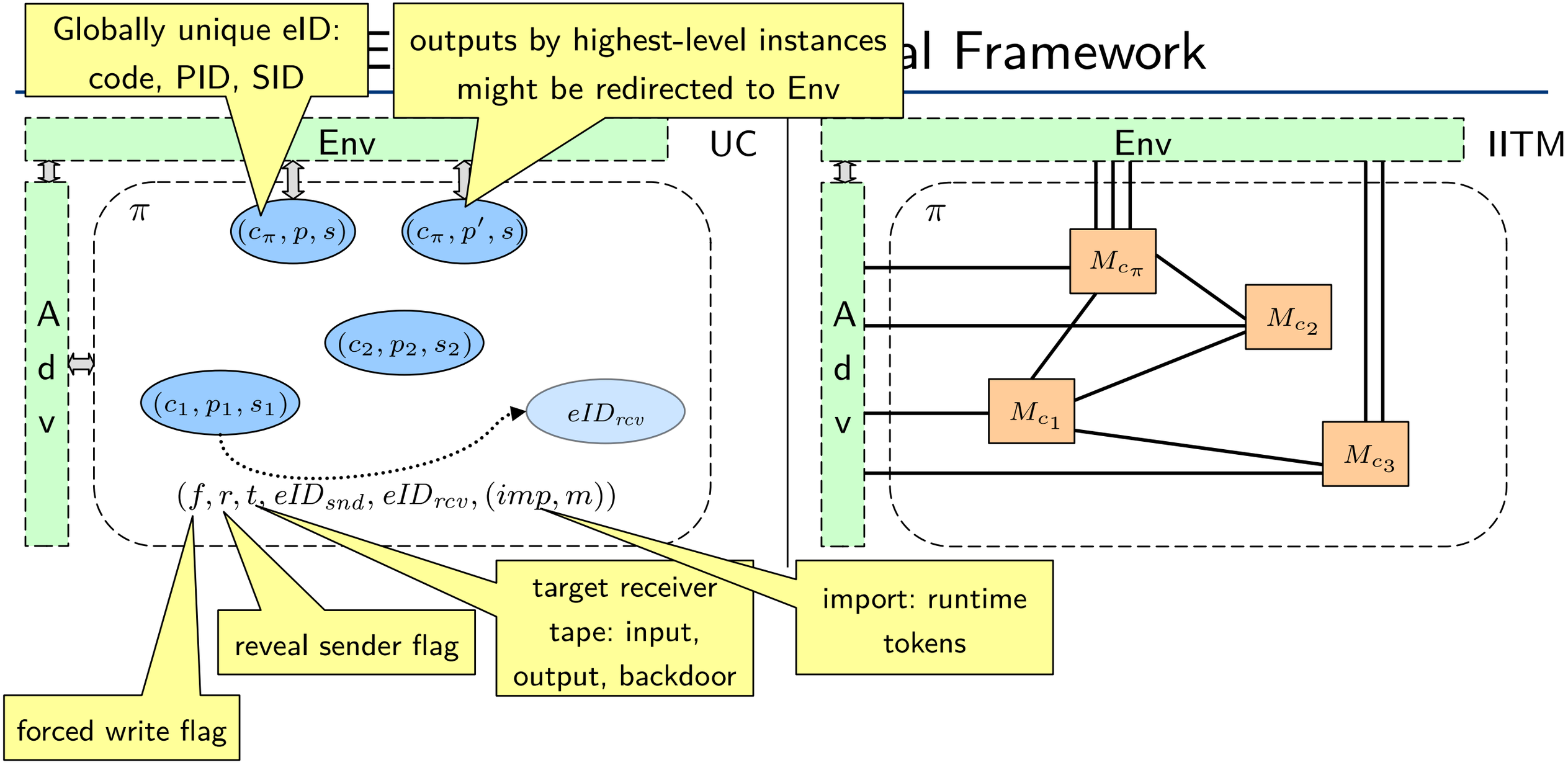
E al Framework



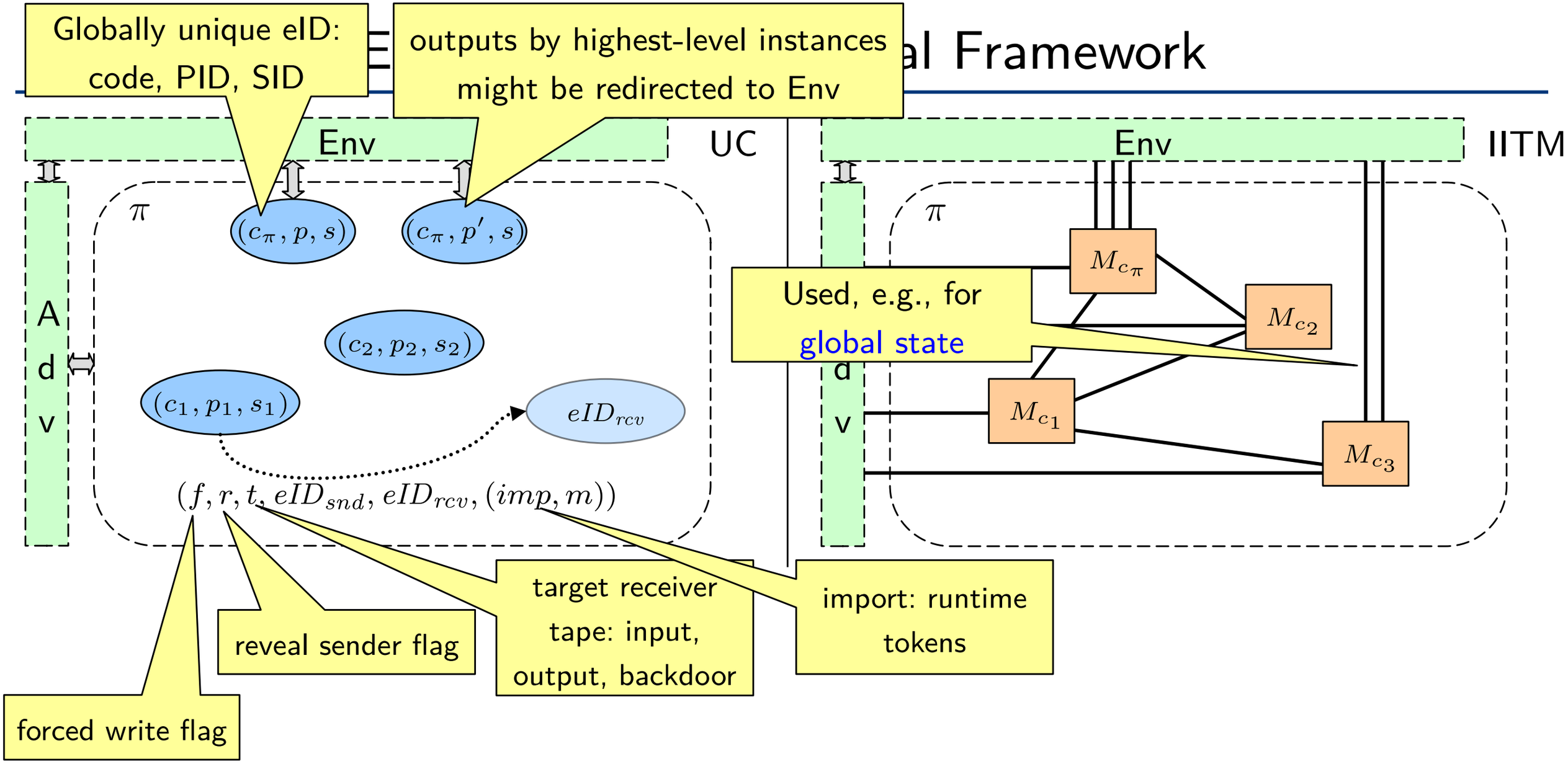
E al Framework



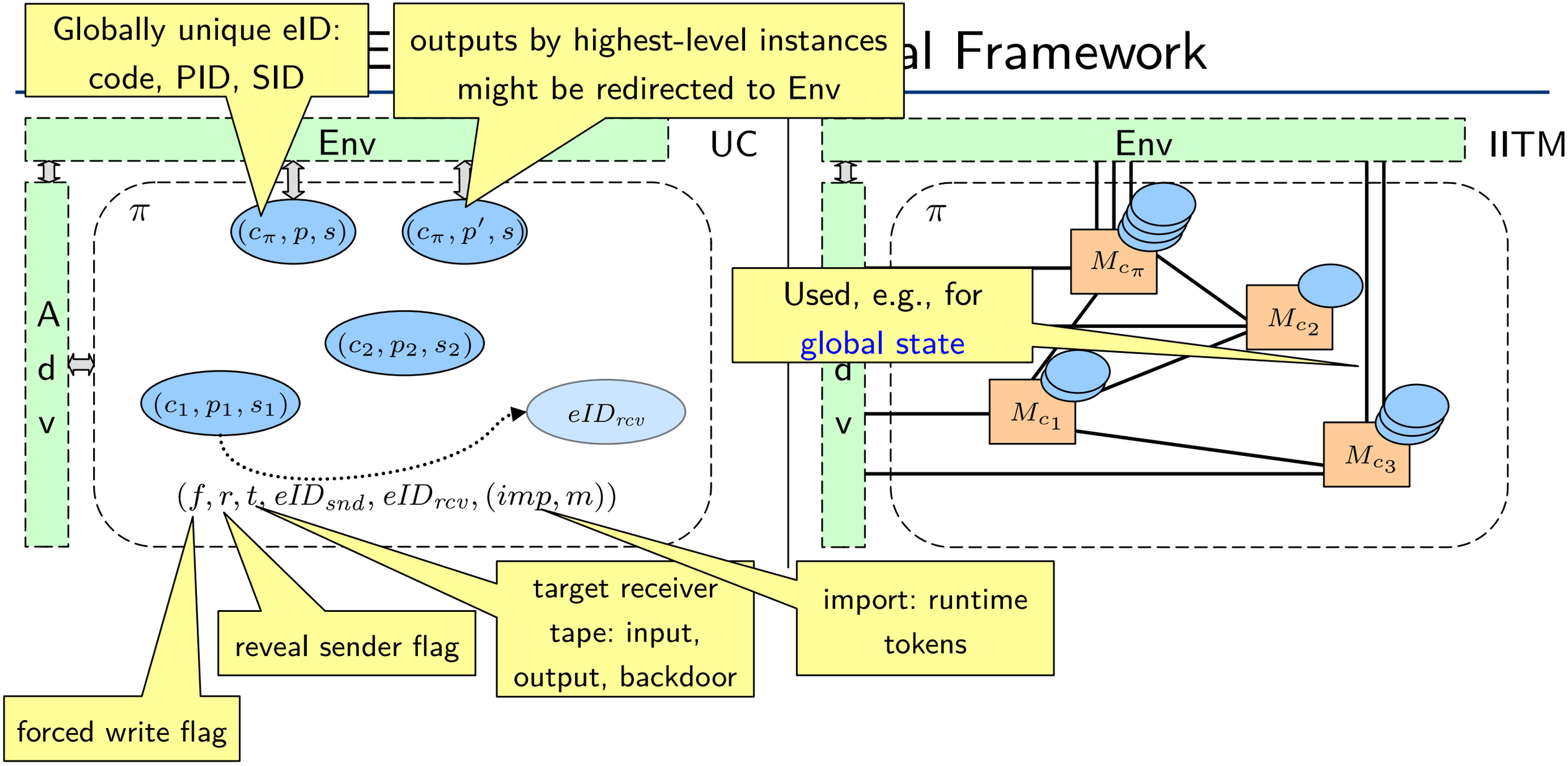
E al Framework



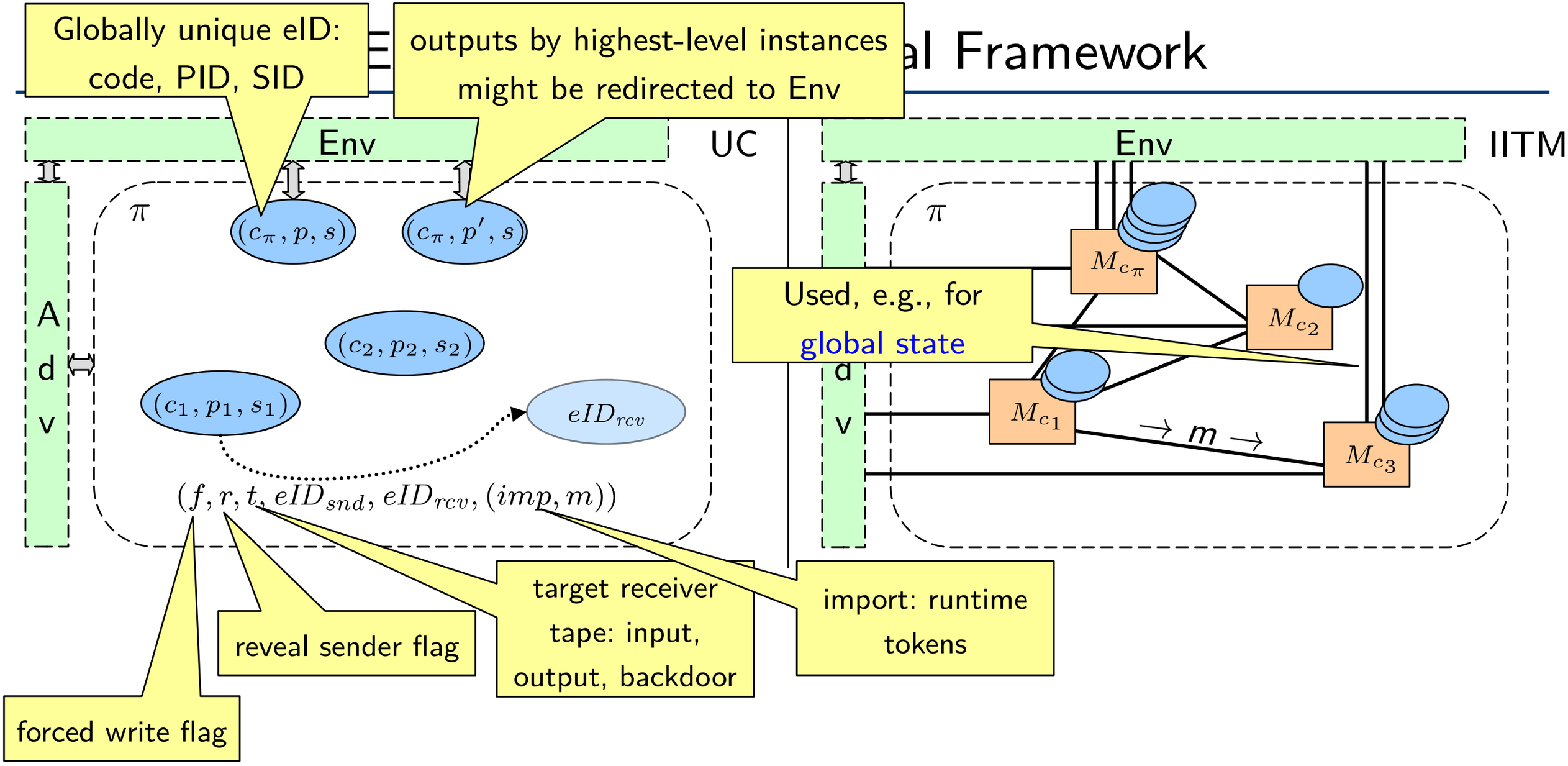
E al Framework



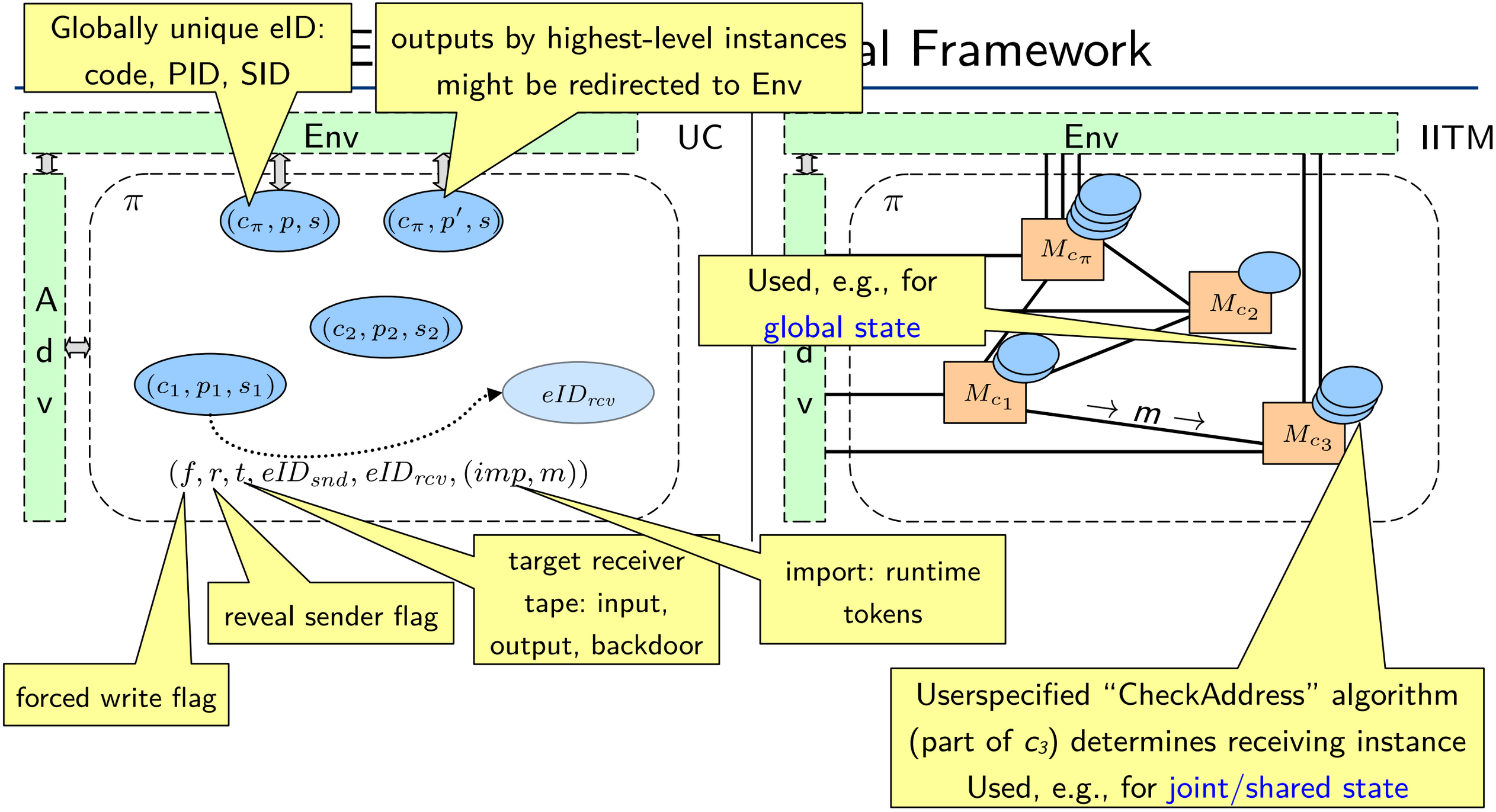
E al Framework



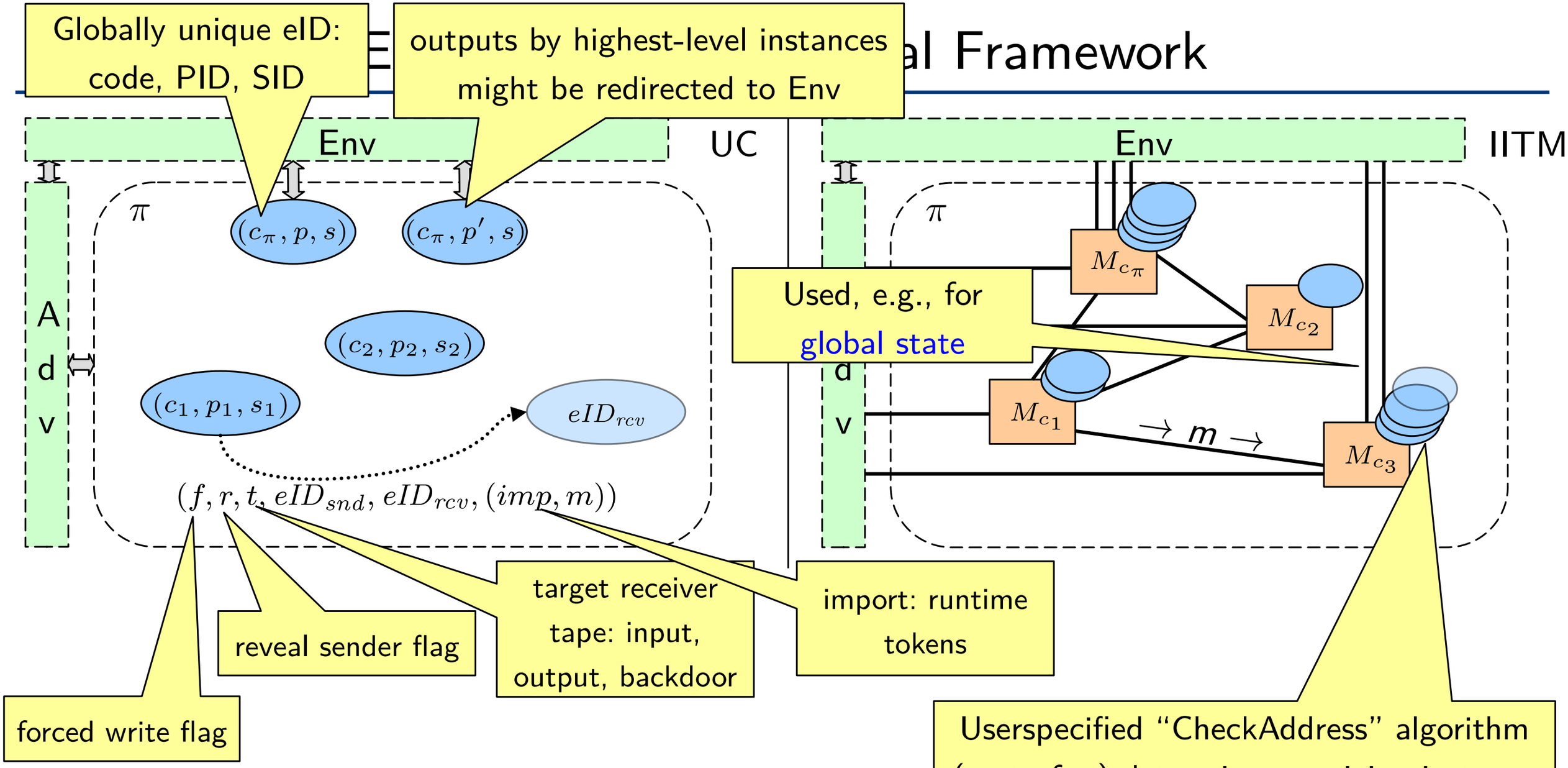
E al Framework



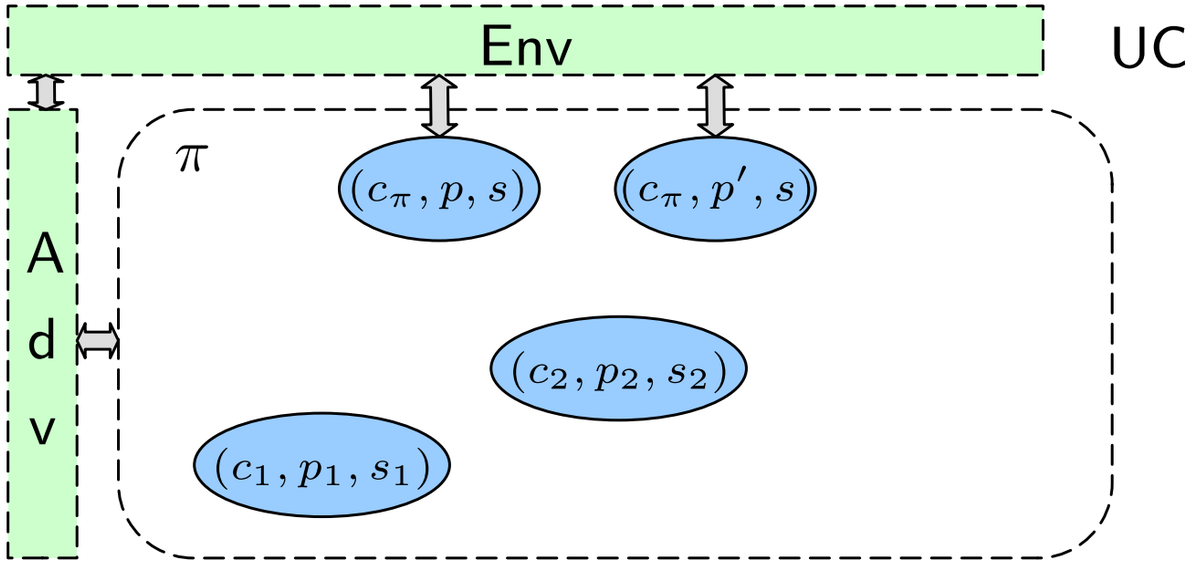
E al Framework



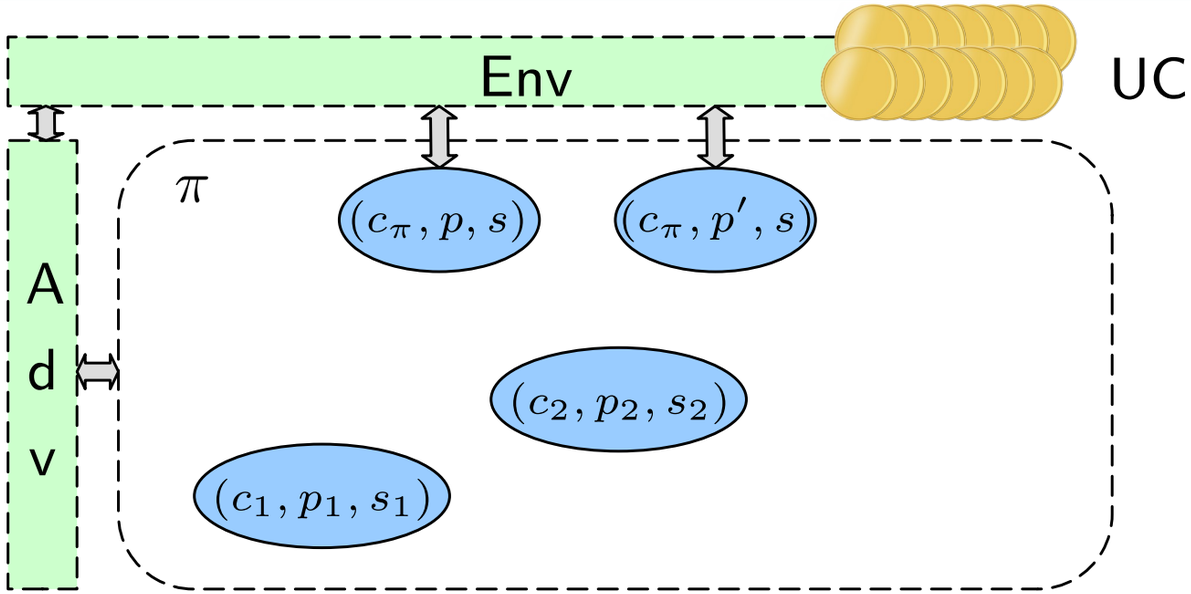
E al Framework



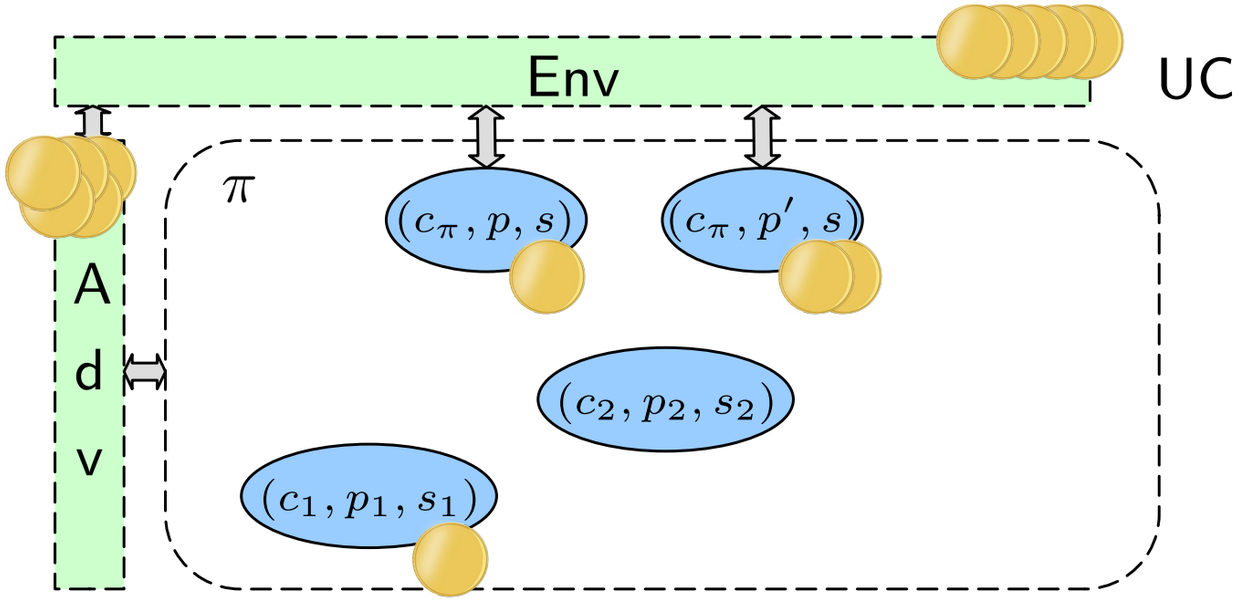
Example: Polynomial Runtime Notion



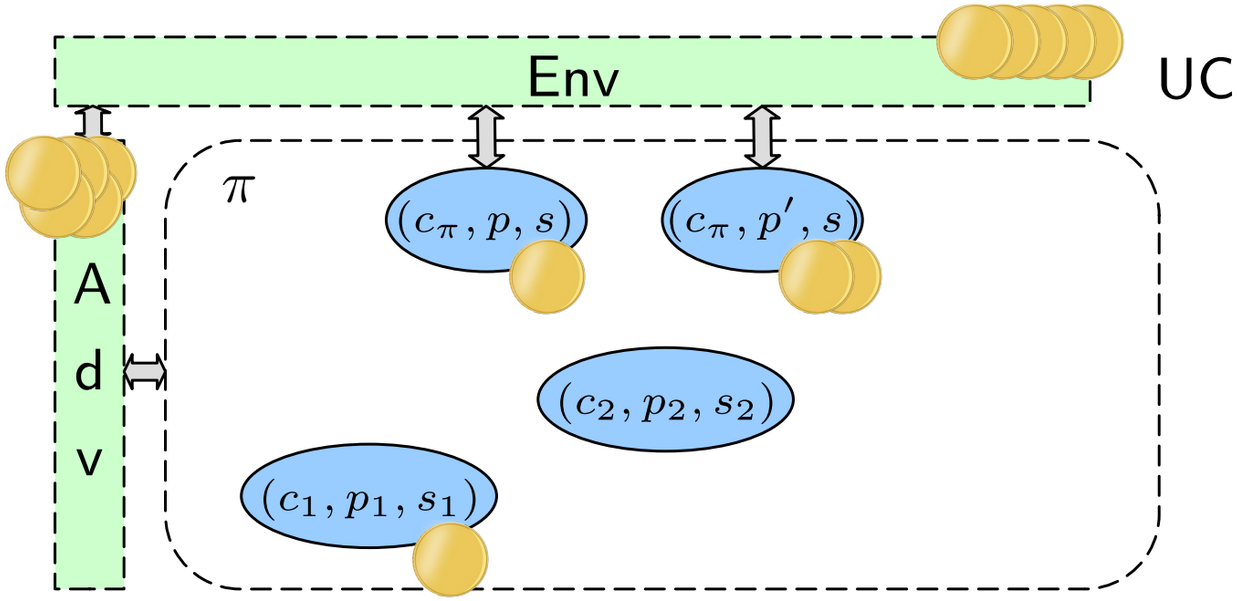
Example: Polynomial Runtime Notion



Example: Polynomial Runtime Notion



Example: Polynomial Runtime Notion

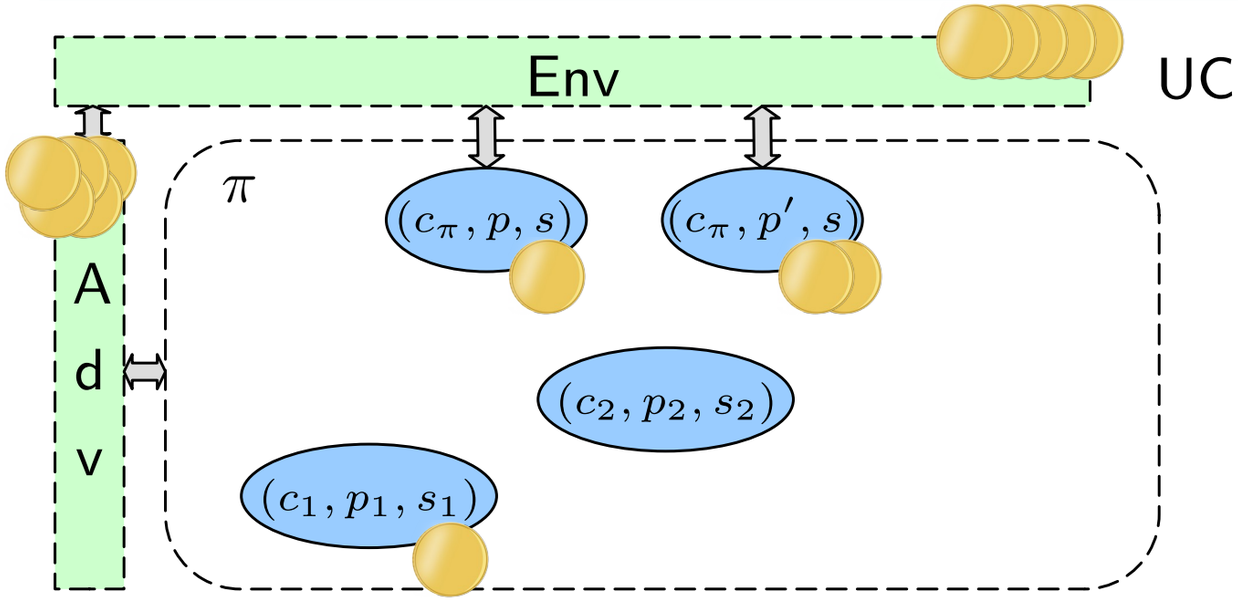


- All instances must be polytime in their currently held runtime tokens ("import")

  → Environment can determine runtime and force protocol/simulator to stop

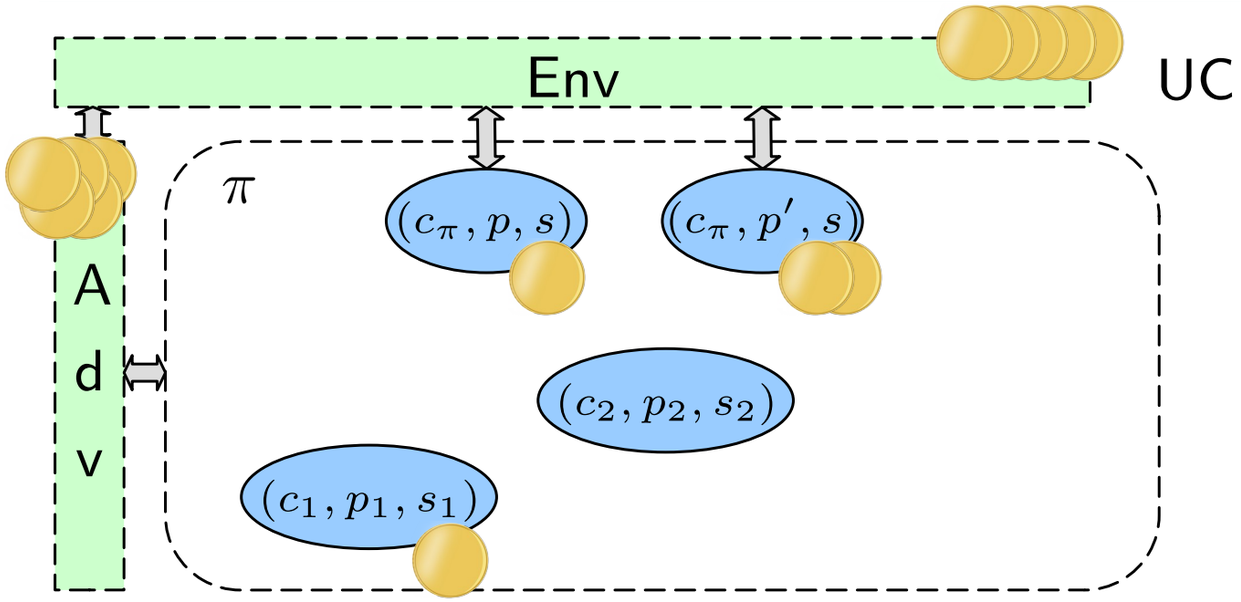
  → Has to be taken into account in simulation

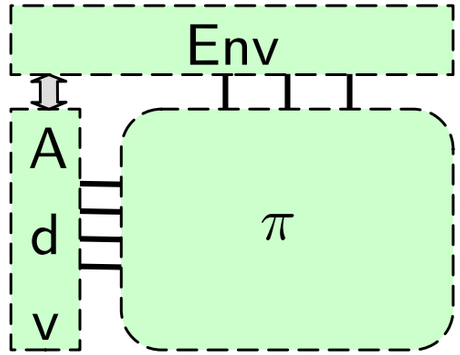
Example: Polynomial Runtime Notion



- All instances must be polytime in their currently held runtime tokens ("import")
  - → Environment can determine runtime and force protocol/simulator to stop
  - → Has to be taken into account in simulation
- Balanced environment: Provide at least as many tokens to adversary as to the protocol

Example: Polynomial Runtime Notion

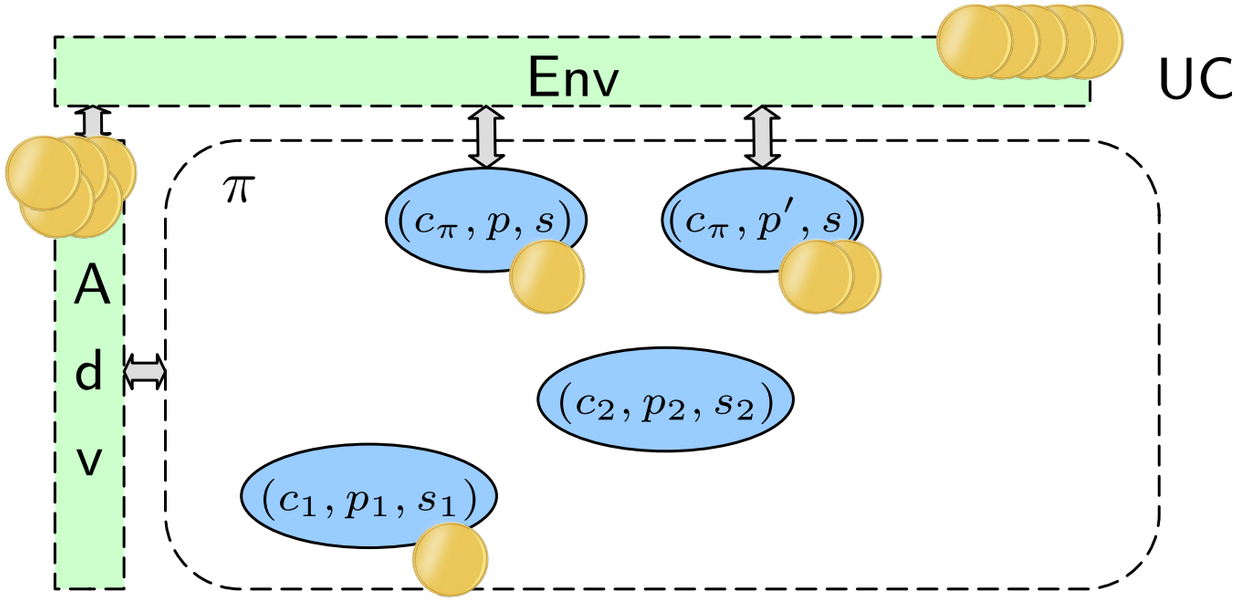




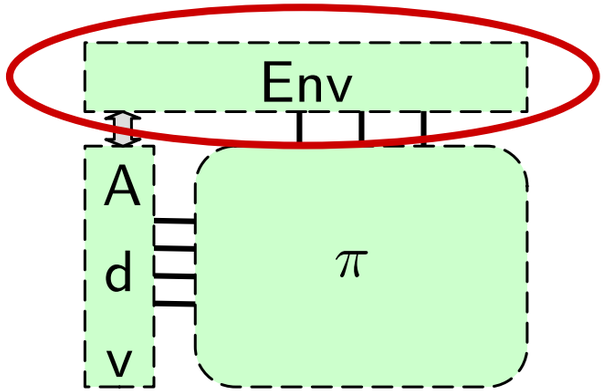
- All instances must be polytime in their currently held runtime tokens ("import")
  - → Environment can determine runtime and force protocol/simulator to stop
  - → Has to be taken into account in simulation
- Balanced environment: Provide at least as many tokens to adversary as to the protocol

runtime notions based on [HUM13]

Example: Polynomial Runtime Notion



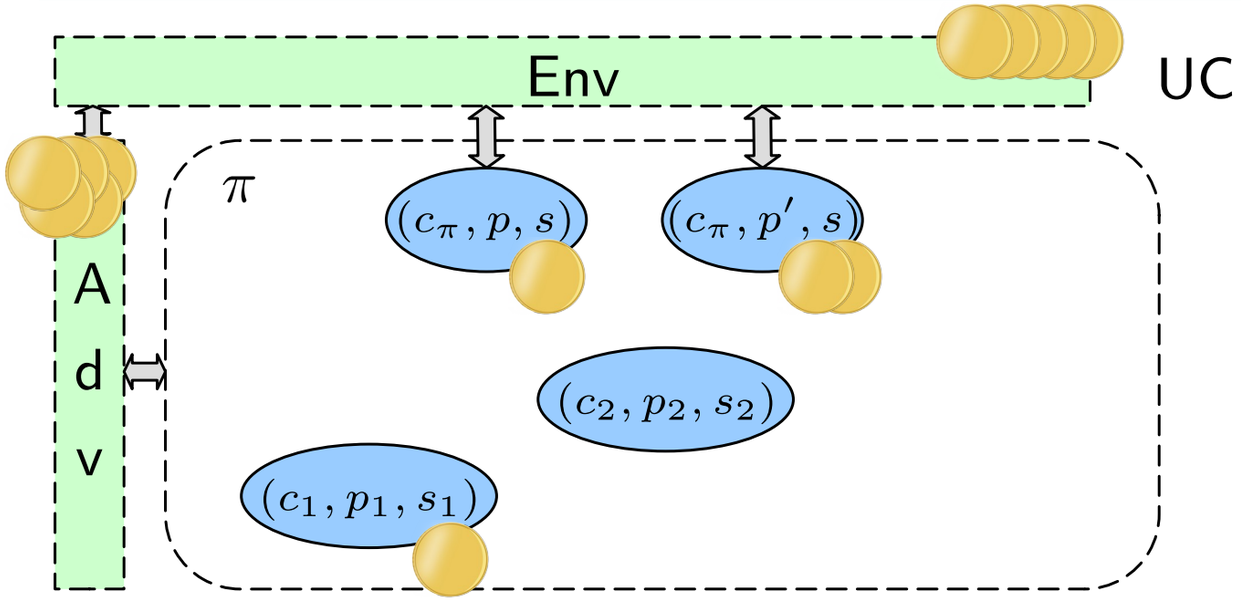
- All instances must be polytime in their currently held runtime tokens ("import")
  - → Environment can determine runtime and force protocol/simulator to stop
  - → Has to be taken into account in simulation
- Balanced environment: Provide at least as many tokens to adversary as to the protocol



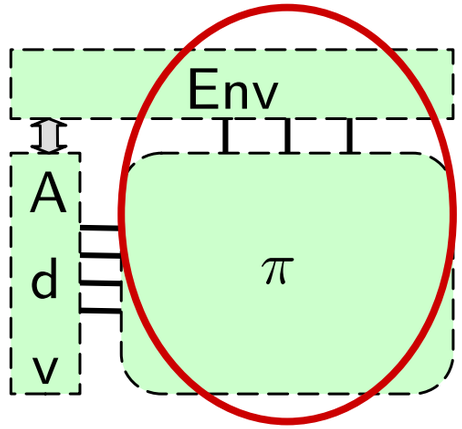
- Environment E: All environments are ppt

runtime notions based on [HUM13]

Example: Polynomial Runtime Notion



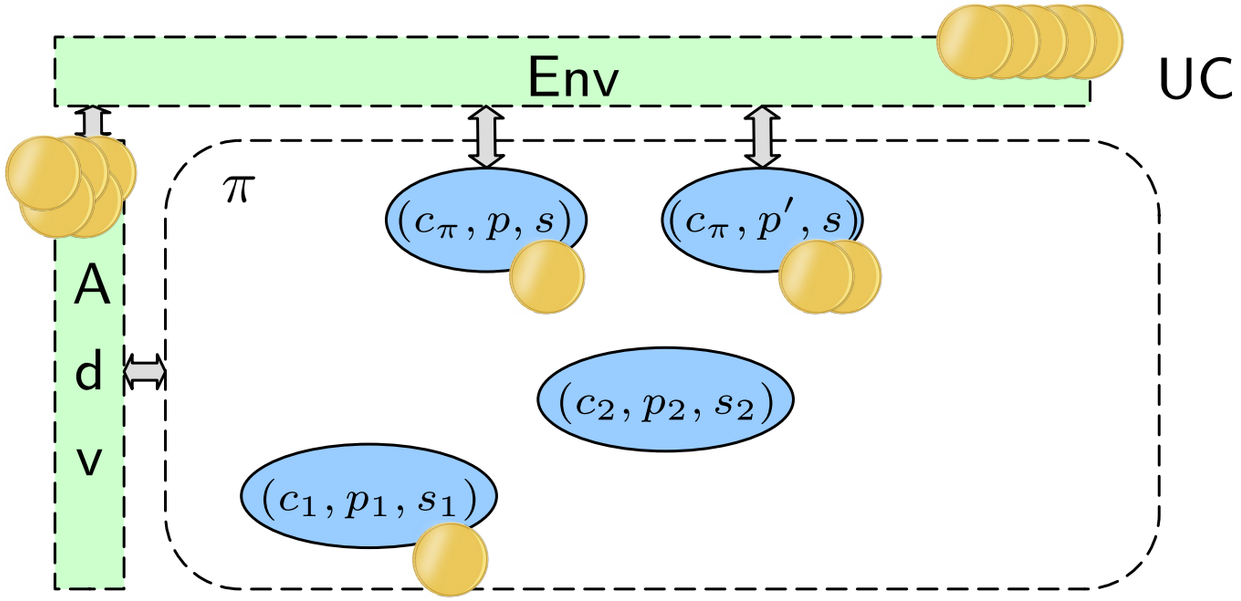
- All instances must be polytime in their currently held runtime tokens ("import")
  - → Environment can determine runtime and force protocol/simulator to stop
  - → Has to be taken into account in simulation
- Balanced environment: Provide at least as many tokens to adversary as to the protocol



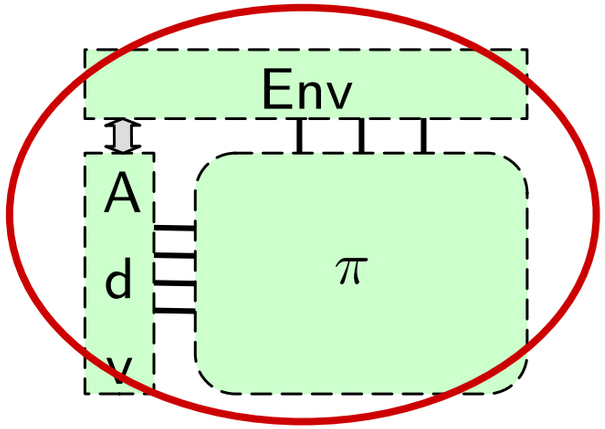
- Environment E: All environments are ppt
- Protocol π: $\forall$ environments E: $(E \mid \pi)$ is overall ppt
  - → Naturally met by protocols from literature

runtime notions based on [HUM13]

Example: Polynomial Runtime Notion



- All instances must be polytime in their currently held runtime tokens ("import")
  - → Environment can determine runtime and force protocol/simulator to stop
  - → Has to be taken into account in simulation
- Balanced environment: Provide at least as many tokens to adversary as to the protocol



- Environment E: All environments are ppt
- Protocol π: $\forall$ environments E: $(E \mid \pi)$ is overall ppt
  - → Naturally met by protocols from literature
- Adv A for protocol π:
  $\forall$ environments E: $(E \mid A \mid \pi)$ is overall ppt

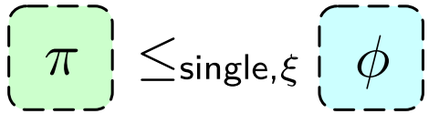
runtime notions based on [HUM13]

Example: Composition

UC Theorem:

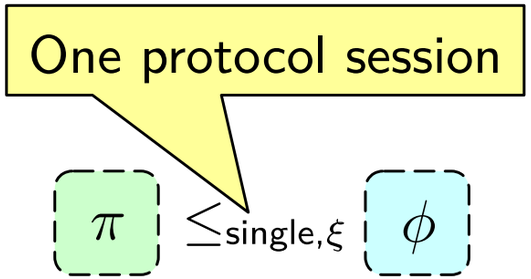
Example: Composition

UC Theorem:



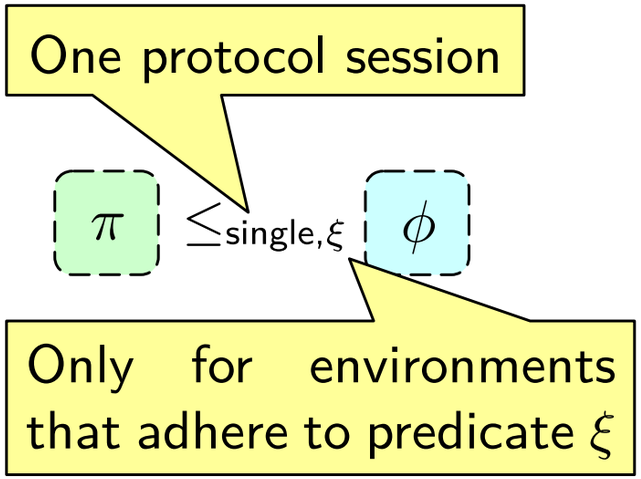
Example: Composition

UC Theorem:



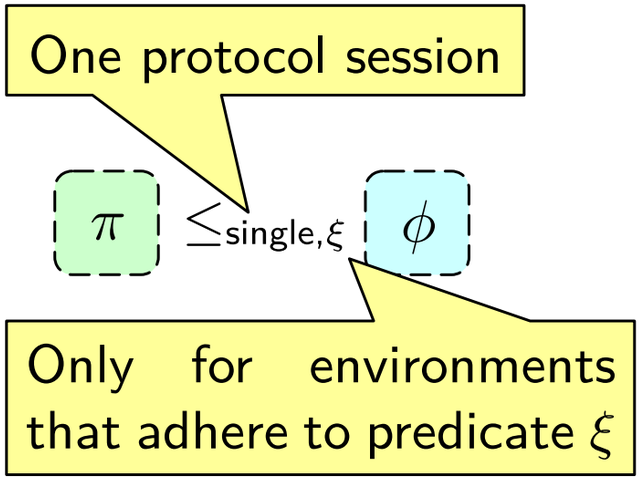
Example: Composition

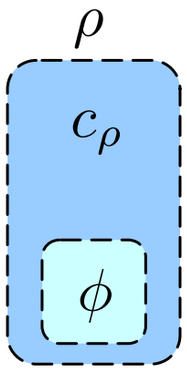
UC Theorem:



Example: Composition

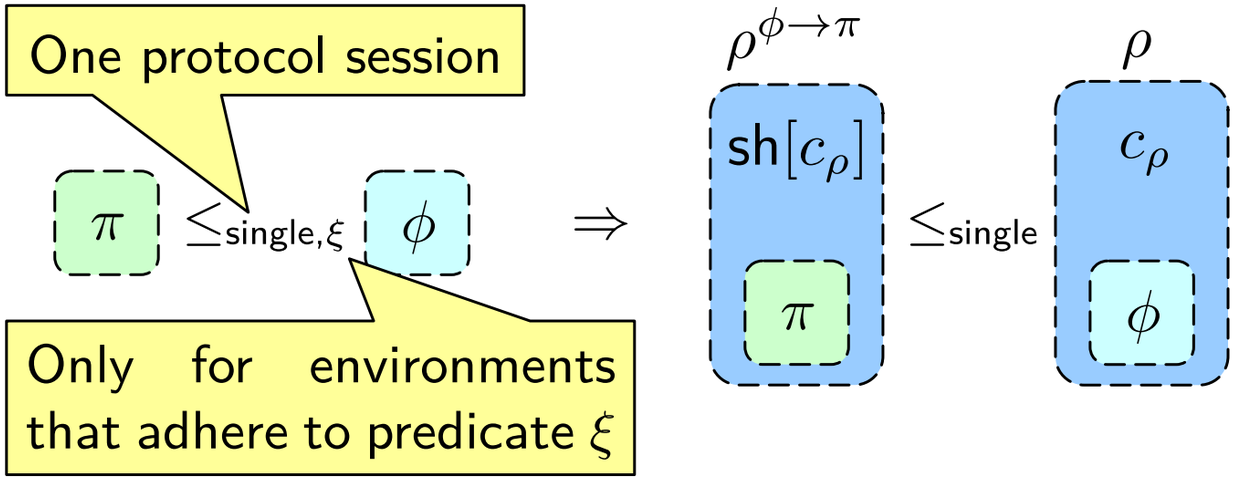
UC Theorem:





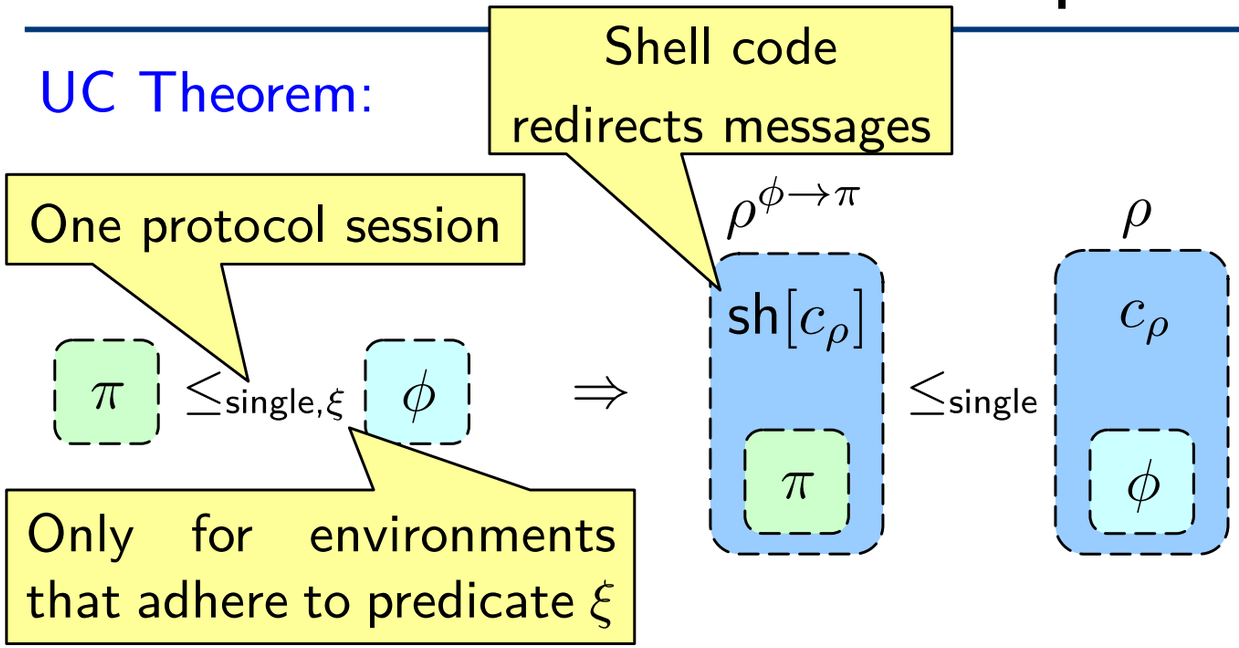
Example: Composition

UC Theorem:



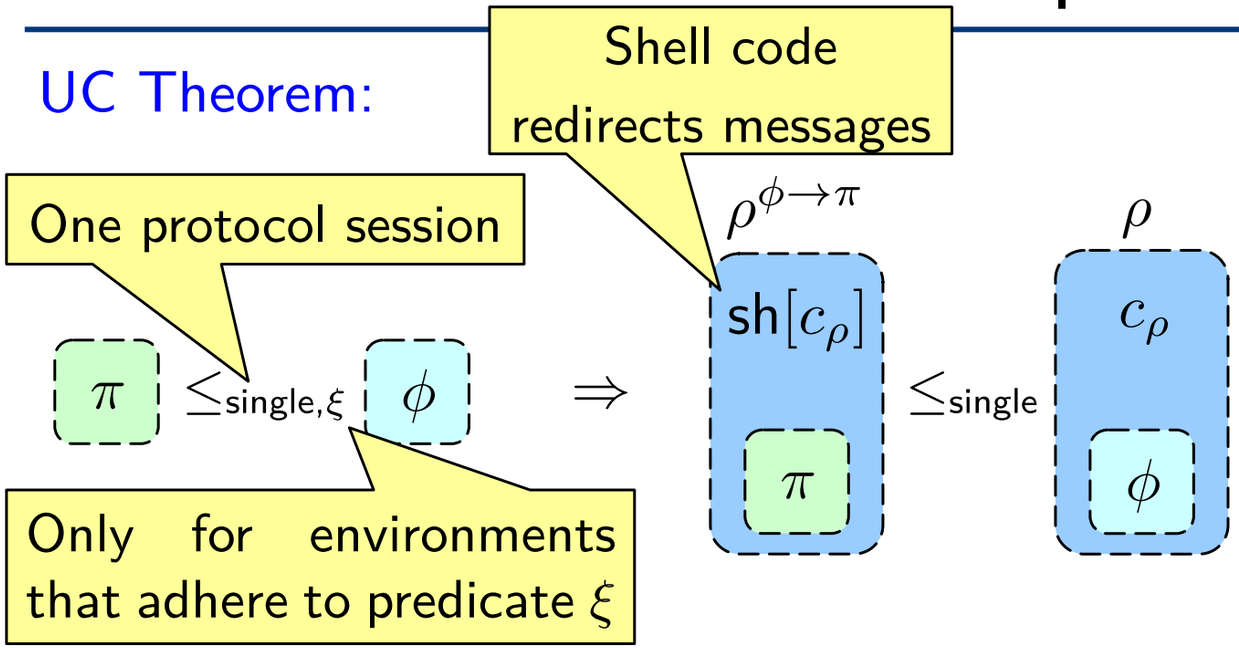
Example: Composition

UC Theorem:



Example: Composition

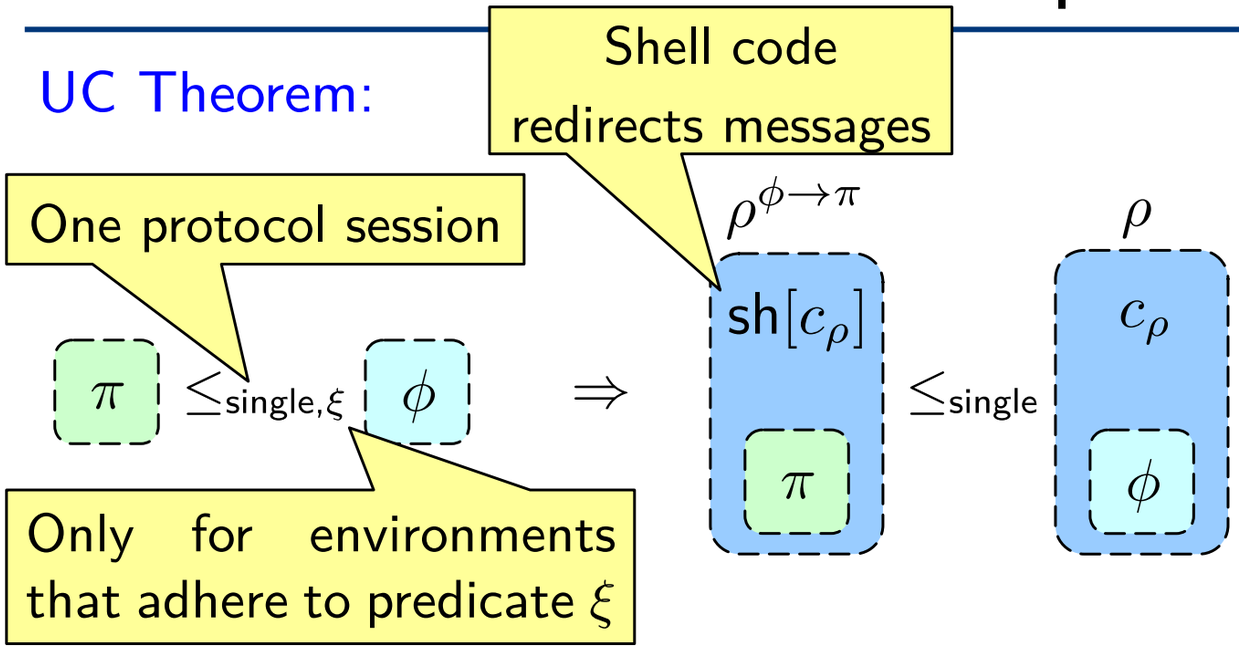
UC Theorem:



- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

Example: Composition

UC Theorem:

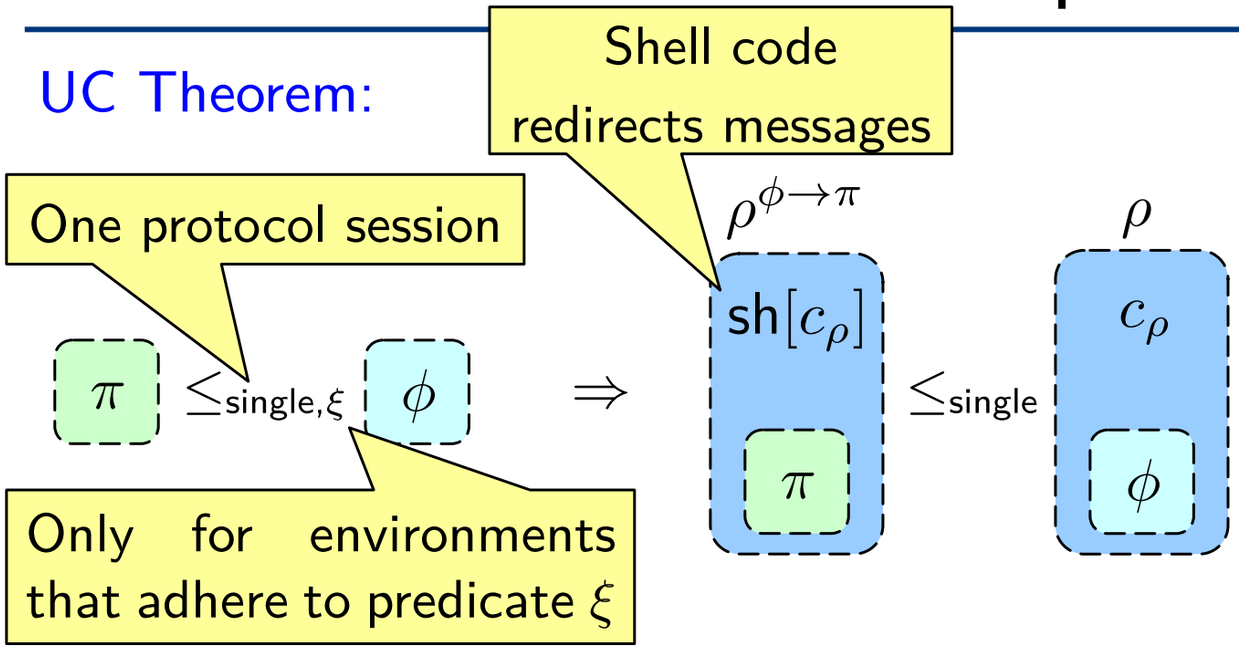


- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

IITM Main Theorem:

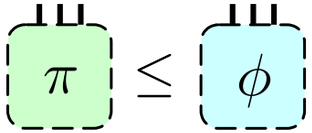
Example: Composition

UC Theorem:



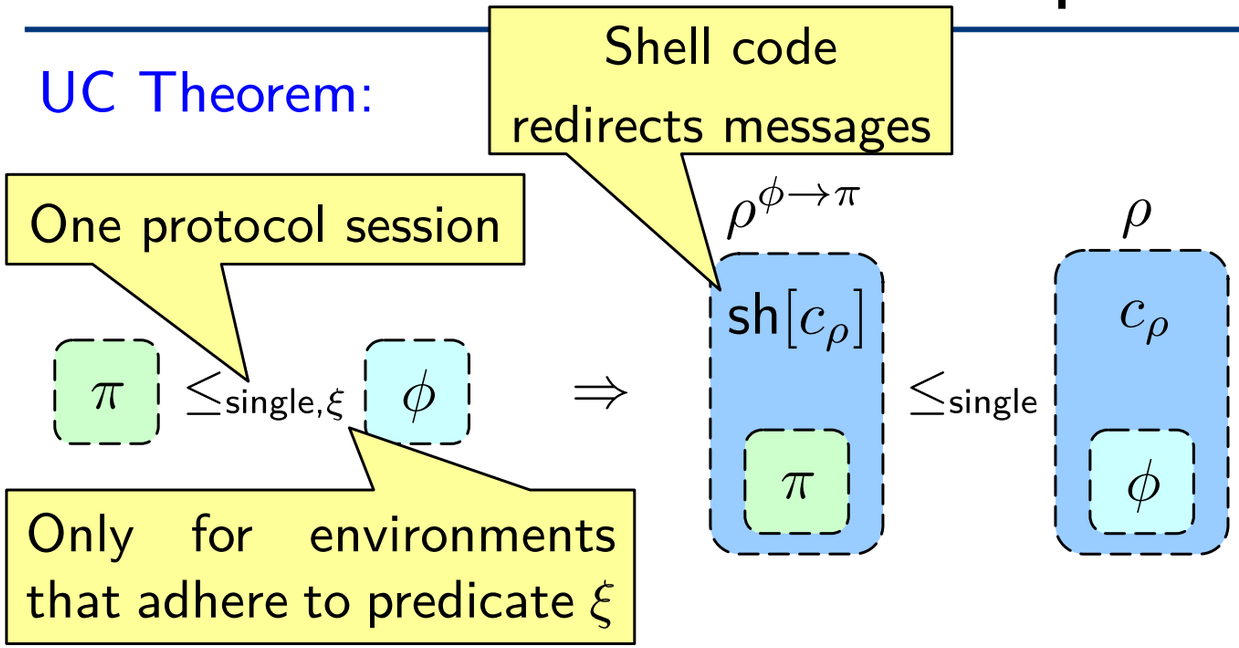
- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

IITM Main Theorem:



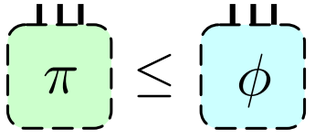
Example: Composition

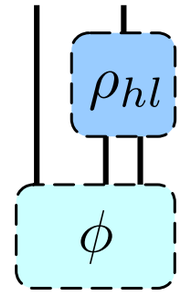
UC Theorem:



- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

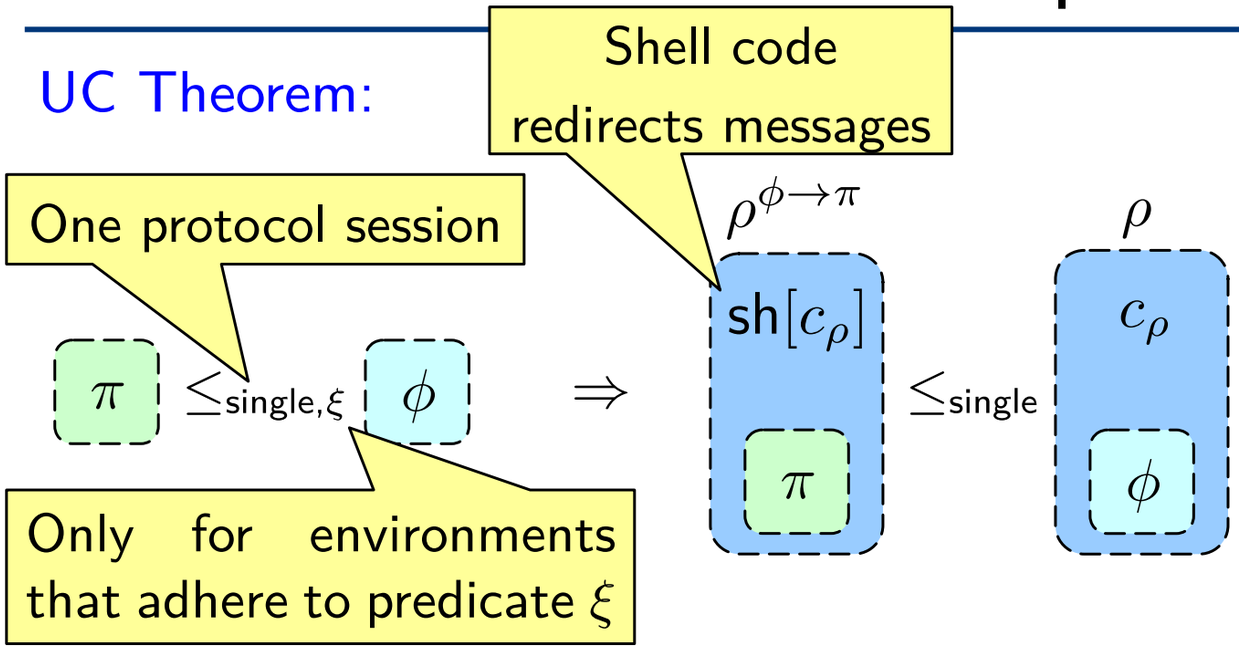
IITM Main Theorem:





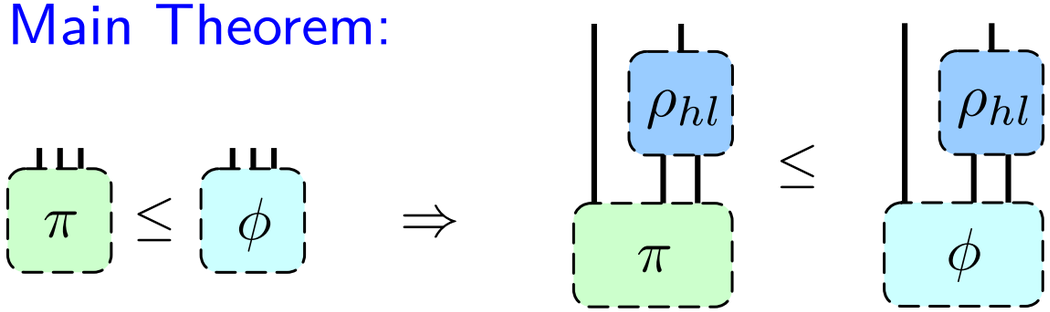
Example: Composition

UC Theorem:



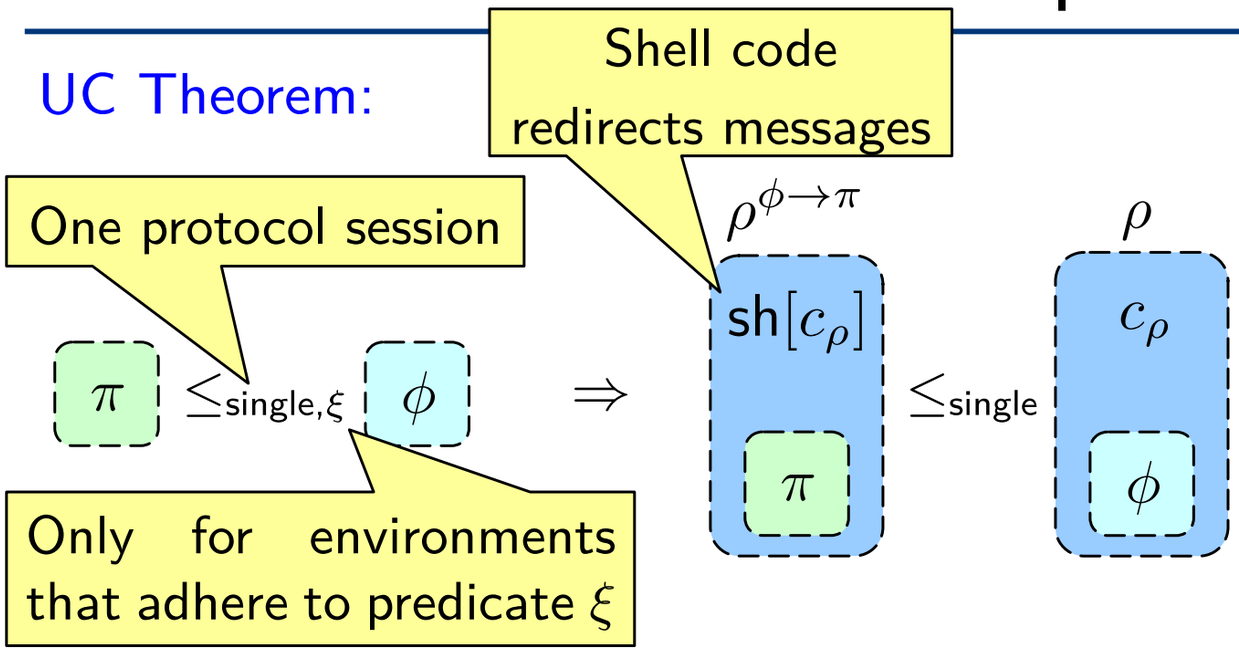
- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

IITM Main Theorem:



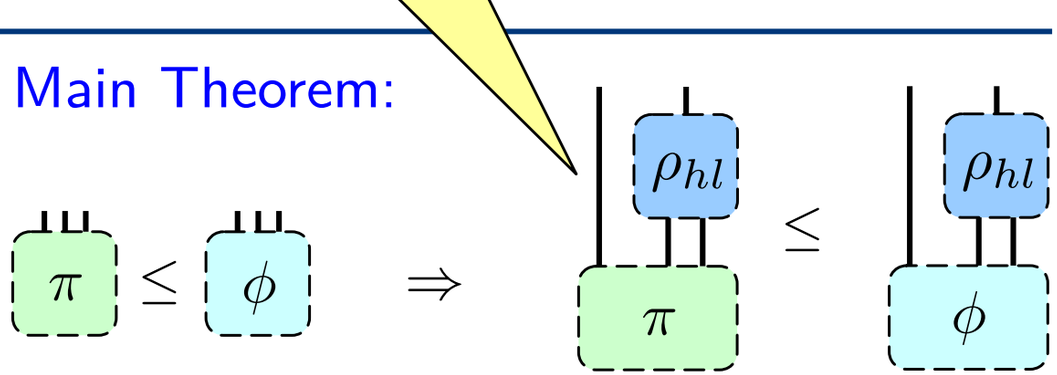
Example: Composition

UC Theorem:



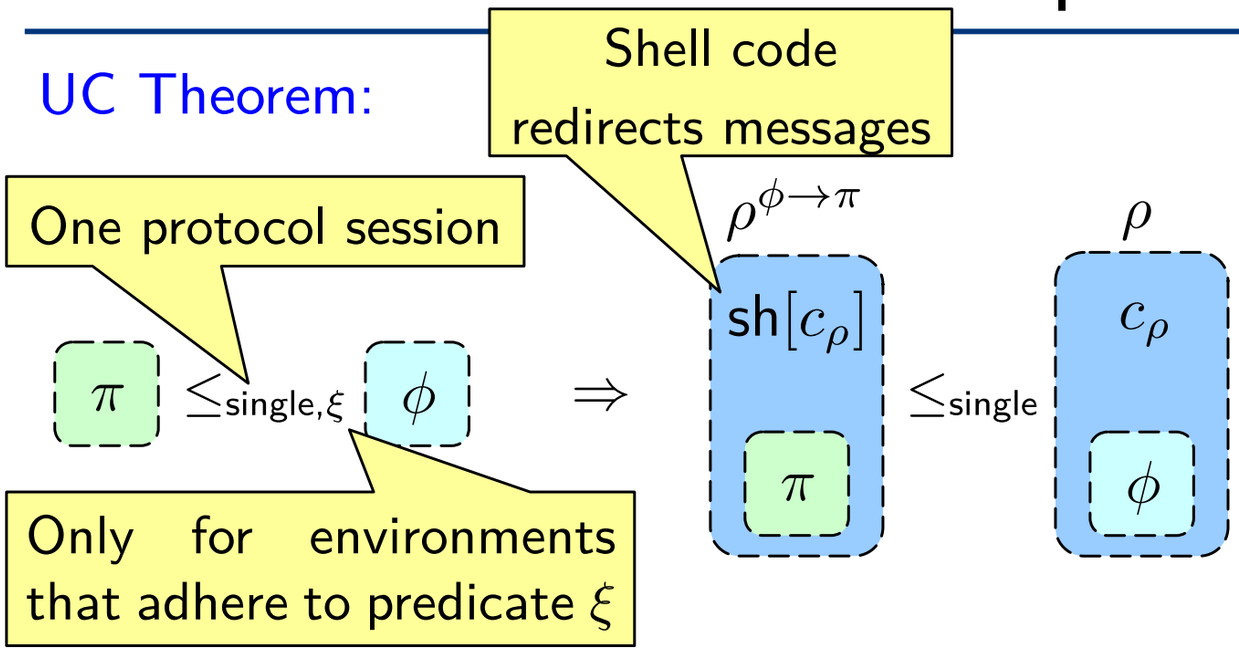
- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

IITM Main Theorem:



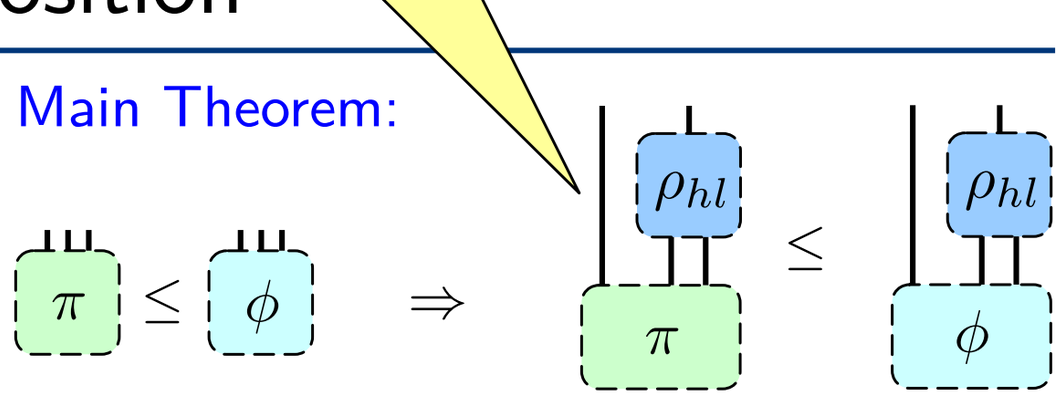
Example: Composition

UC Theorem:



- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

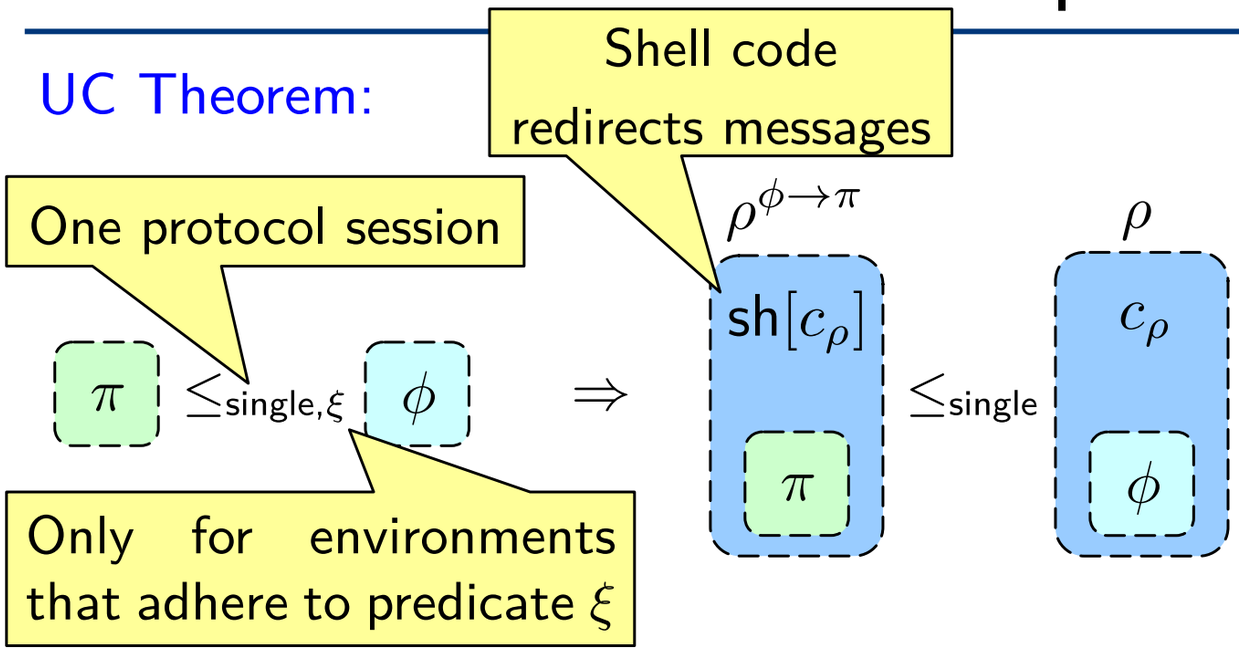
IITM Main Theorem:



- ρ_{hl} only connects to external tapes of π/ϕ

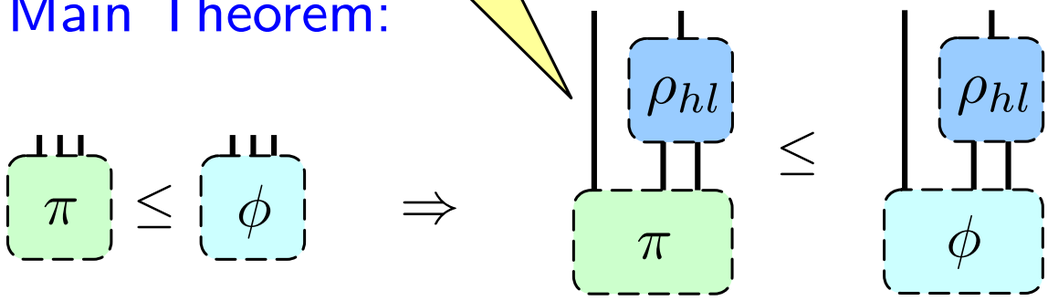
Example: Composition

UC Theorem:



- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

IITM Main Theorem:

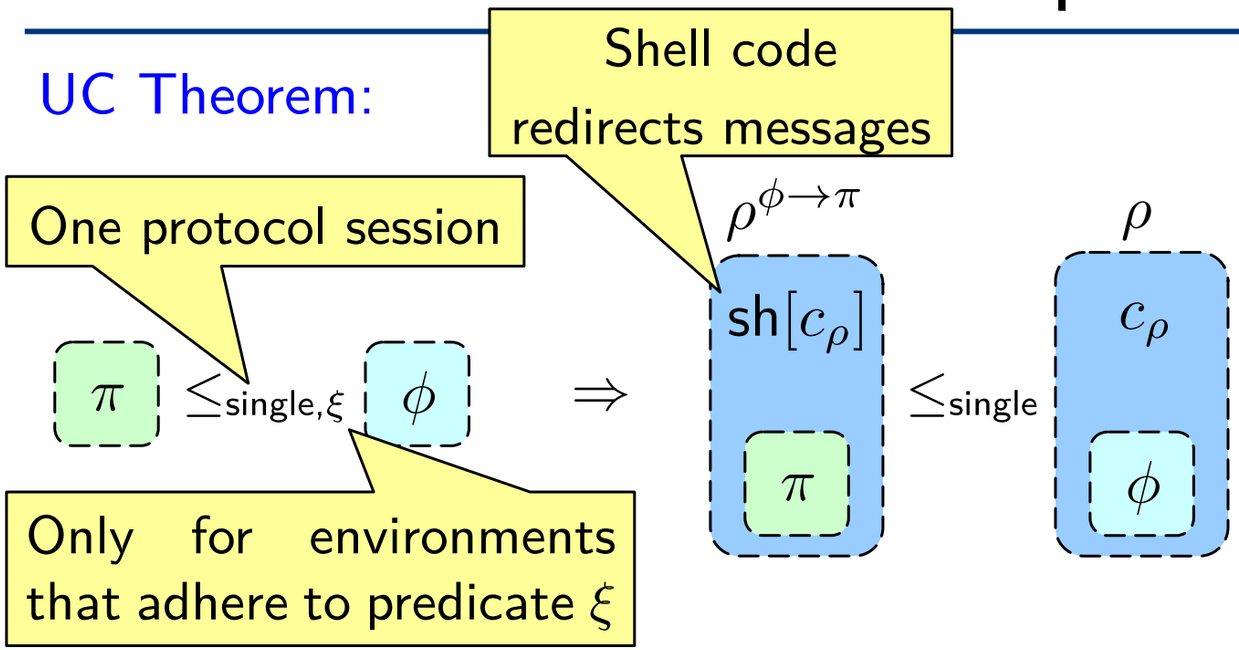


- ρ_{hl} only connects to external tapes of π/ϕ

Multi session setting: Includes protocols with joint, shared, and global state as special cases

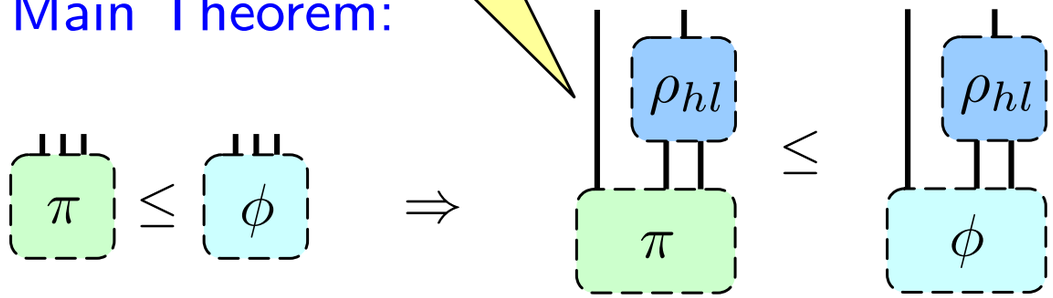
Example: Composition

UC Theorem:



- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

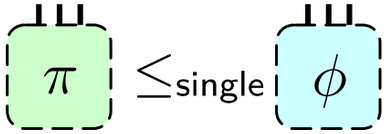
IITM Main Theorem:



- ρ_{hl} only connects to external tapes of π/ϕ

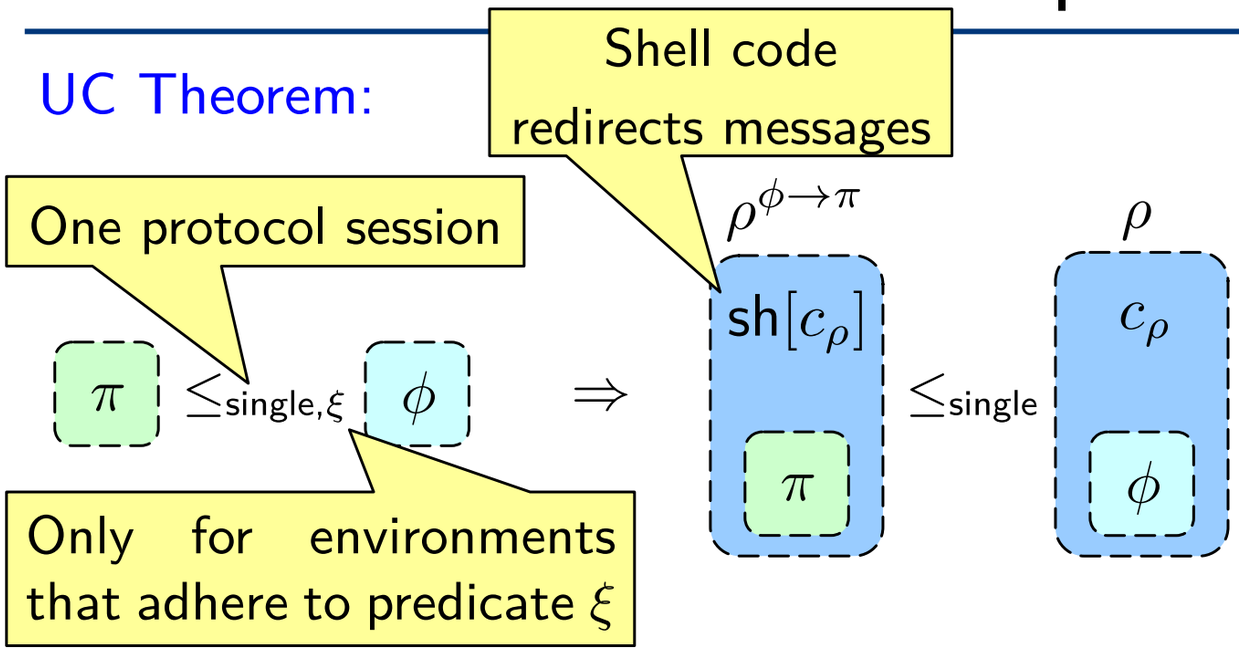
Multi session setting: Includes protocols with joint, shared, and global state as special cases

IITM Second Theorem:



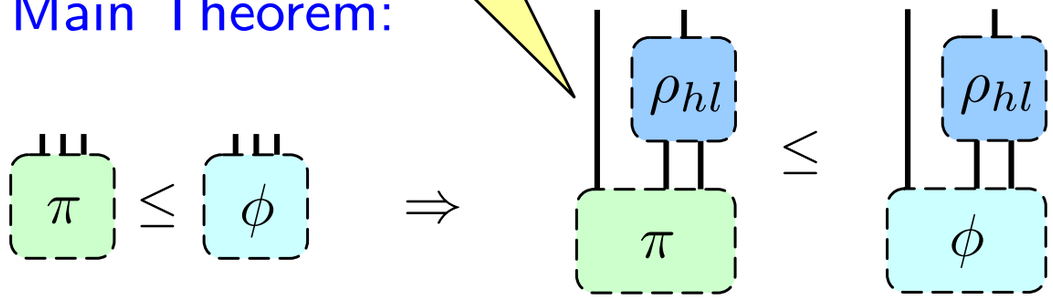
Example: Composition

UC Theorem:



- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

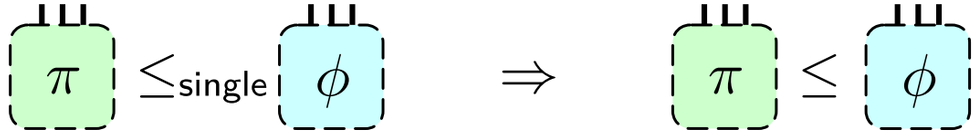
IITM Main Theorem:



- ρ_{hl} only connects to external tapes of π/ϕ

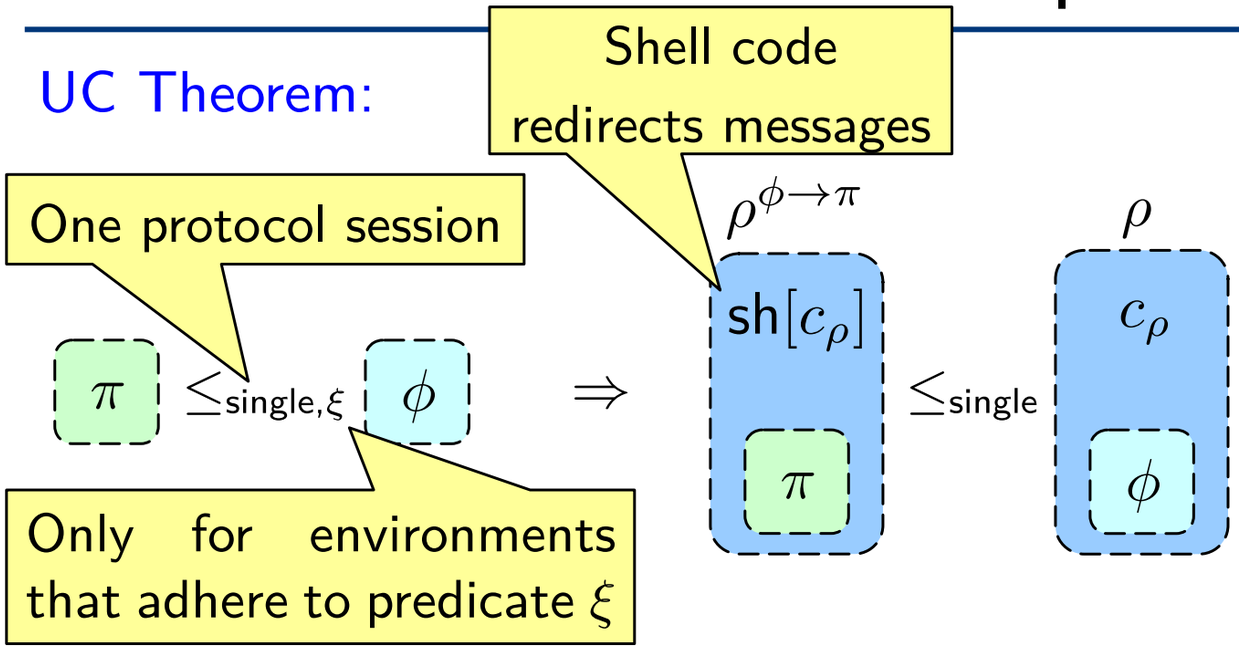
Multi session setting: Includes protocols with joint, shared, and global state as special cases

IITM Second Theorem:



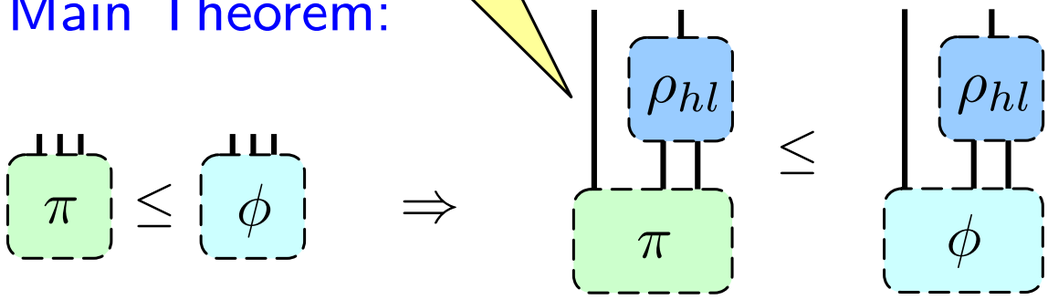
Example: Composition

UC Theorem:



- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

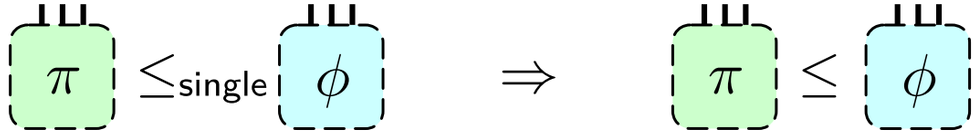
IITM Main Theorem:



- ρ_{hl} only connects to external tapes of π/ϕ

Multi session setting: Includes protocols with joint, shared, and global state as special cases

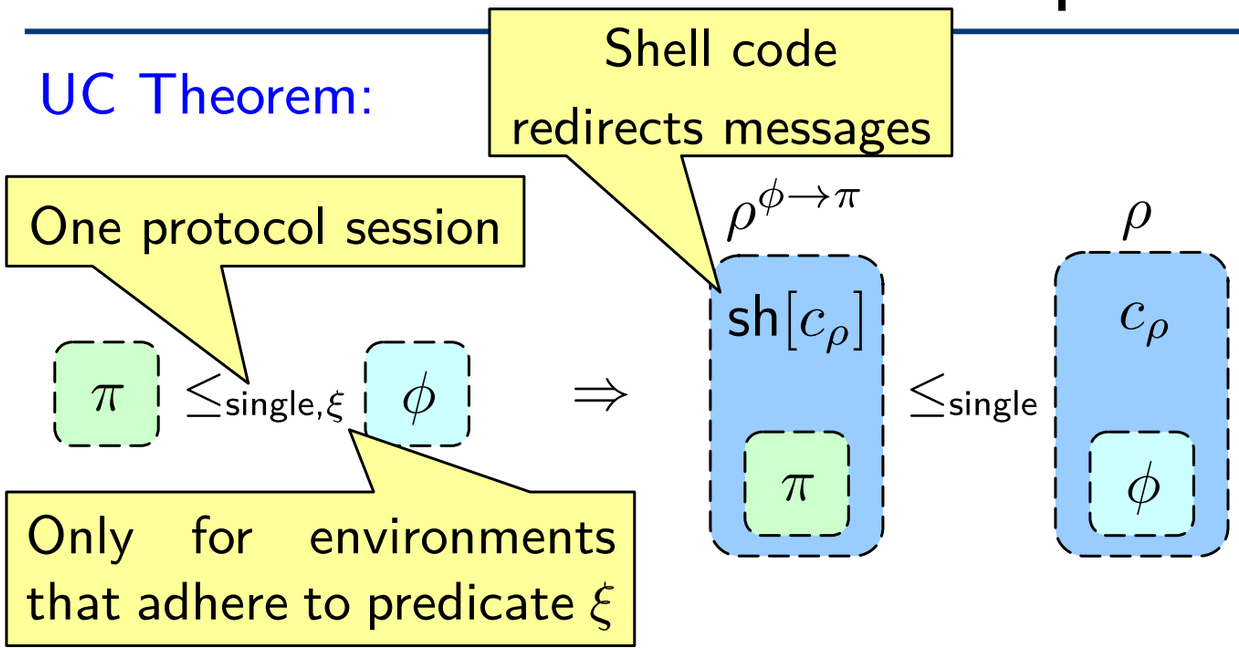
IITM Second Theorem:



- π/ϕ are σ-session protocols

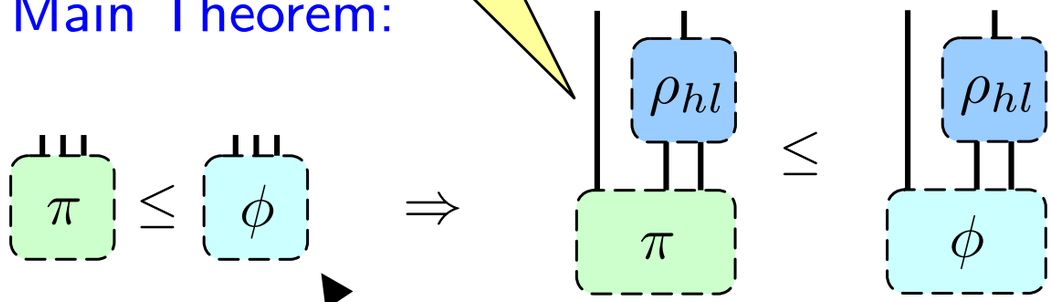
Example: Composition

UC Theorem:



- π/ϕ are subroutine respecting
- π/ϕ are subroutine exposing
- ρ is compliant

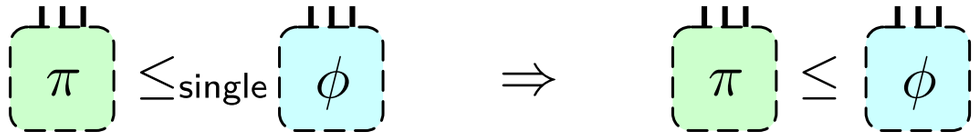
IITM Main Theorem:



- ρ_{hl} only connects to external tapes of π/ϕ

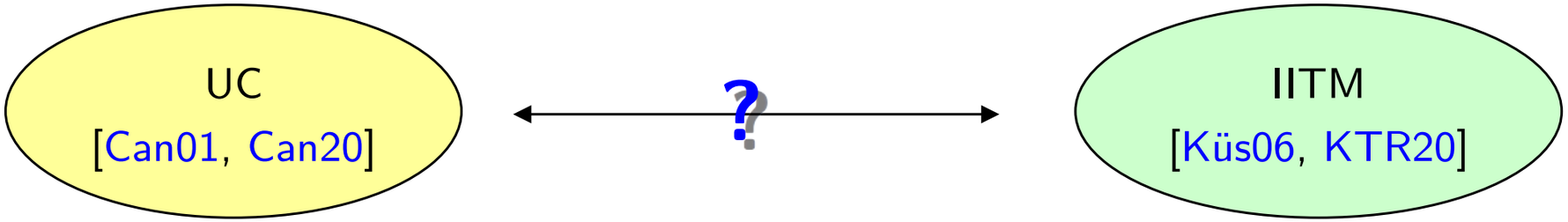
Multi session setting: Includes protocols with joint, shared, and global state as special cases

IITM Second Theorem:

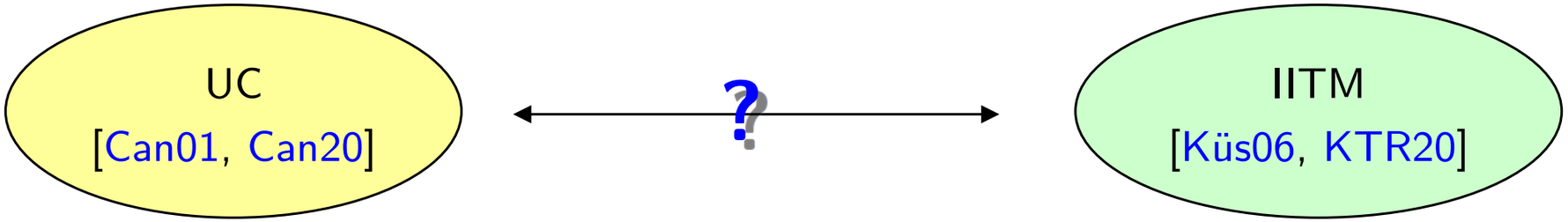


- π/ϕ are σ-session protocols

Relationship?



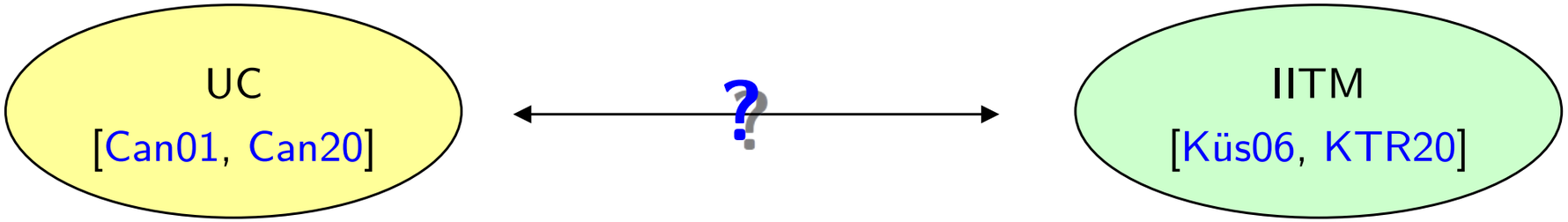
Relationship?



Different...

- computational frameworks
- compositional statements
- classes of environments, adversaries/simulators, protocols

Relationship?



Different...

- computational frameworks
- compositional statements
- classes of environments, adversaries/simulators, protocols

How do models relate?

Are protocols and security results even comparable?

Main Contributions

- Concepts of UC and IITM
- **Embedding UC into IITM**
- The Other Direction

From UC to IITM Protocols

From UC to IITM Protocols

- All aspects of π_{UC} can be translated naturally to an IITM protocol, including dynamic code via Universal Turing machine.

From UC to IITM Protocols

- All aspects of π_{UC} can be translated naturally to an IITM protocol, including dynamic code via Universal Turing machine.

- π_{IITM} has to reveal (upper bound of) runtime tokens received so far to the adversary

  $\rightarrow$ Captures side channel provided by UC environments

From UC to IITM Protocols

- All aspects of π_{UC} can be translated naturally to an IITM protocol, including dynamic code via Universal Turing machine.

- π_{IITM} has to reveal (upper bound of) runtime tokens received so far to the adversary

  $\rightarrow$ Captures side channel provided by UC environments

- Variant $\pi_{IITM}^{\xi\text{-id}}$ that enforces a ξ bound on arbitrary environments

Preserving Security

Theorem:

$$\pi_{\mathsf{UC}} \leq_{\mathsf{single},\xi} \phi_{\mathsf{UC}}$$

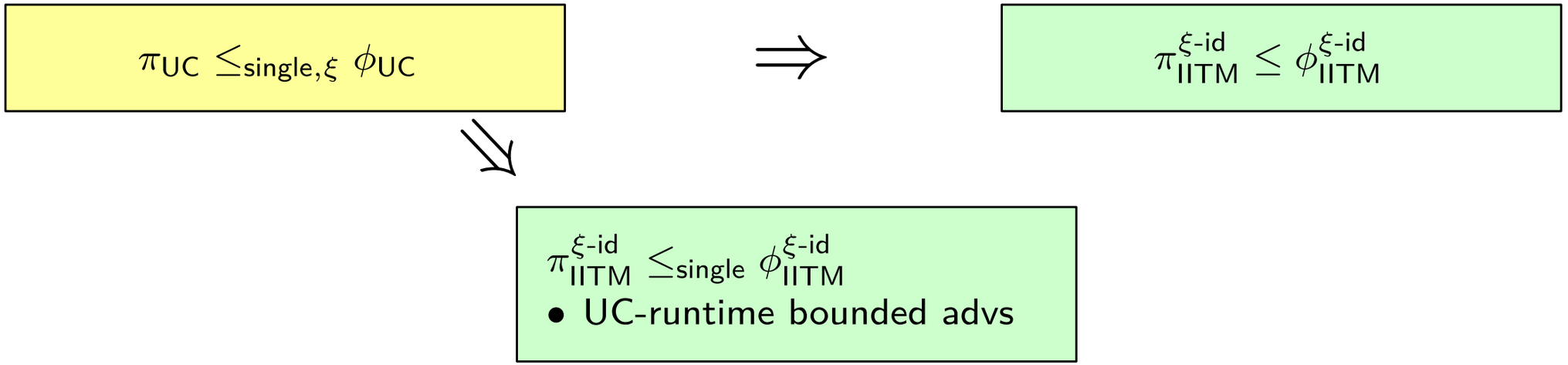
Preserving Security

Theorem:

$$\pi_{\mathsf{UC}} \leq_{\mathsf{single},\xi} \phi_{\mathsf{UC}} \quad \Rightarrow \quad \pi^{\xi\text{-id}}_{\mathsf{IITM}} \leq \phi^{\xi\text{-id}}_{\mathsf{IITM}}$$

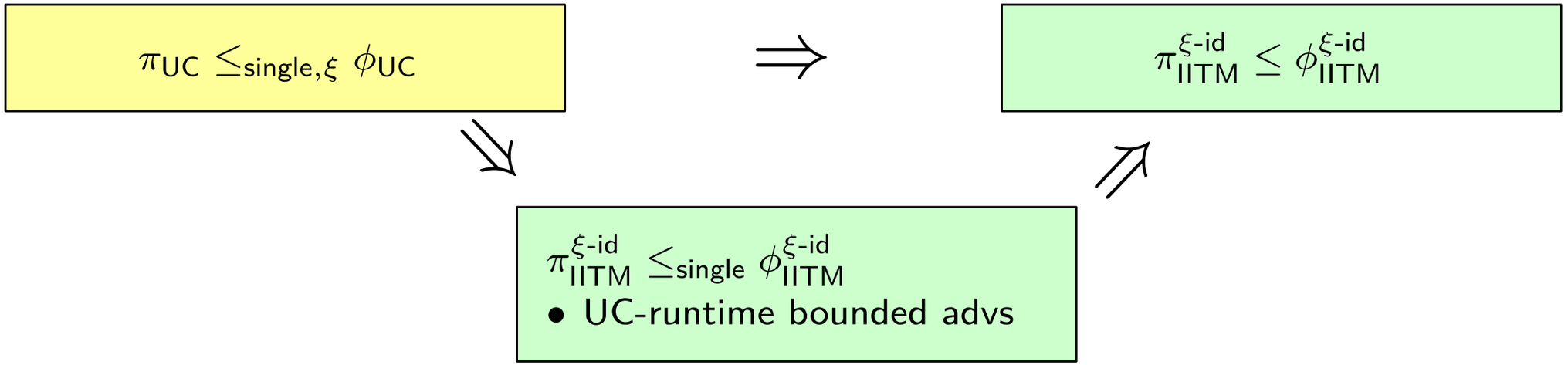
Preserving Security

Theorem:



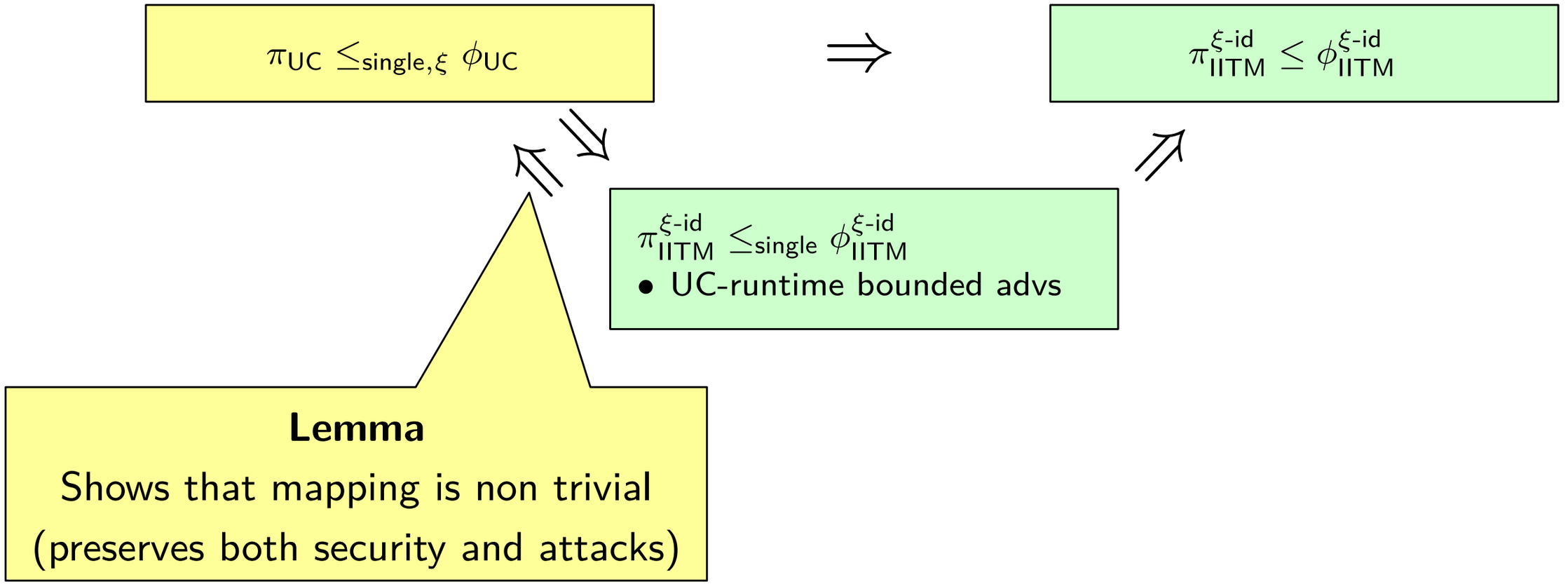
Preserving Security

Theorem:



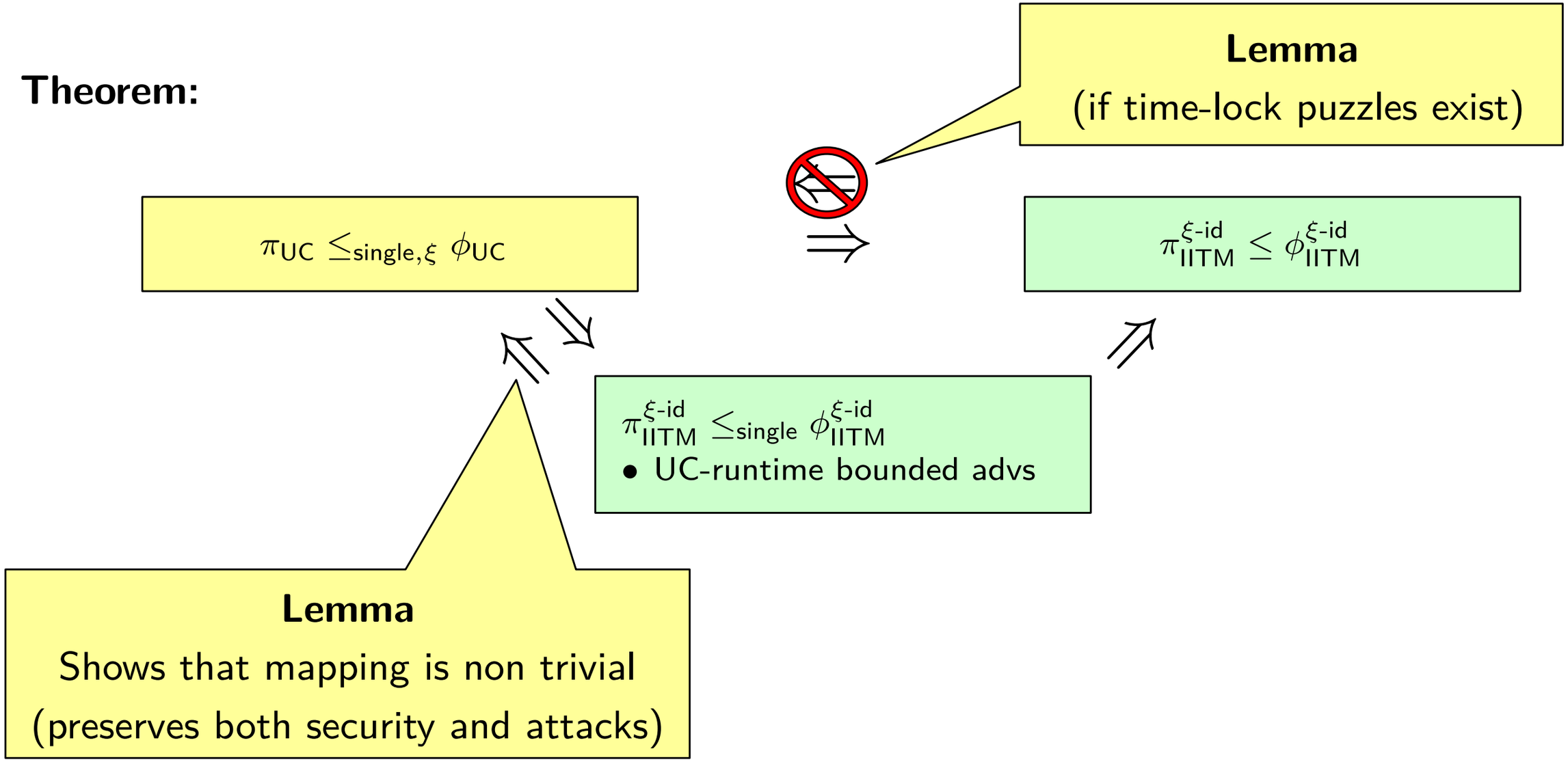
Preserving Security

Theorem:



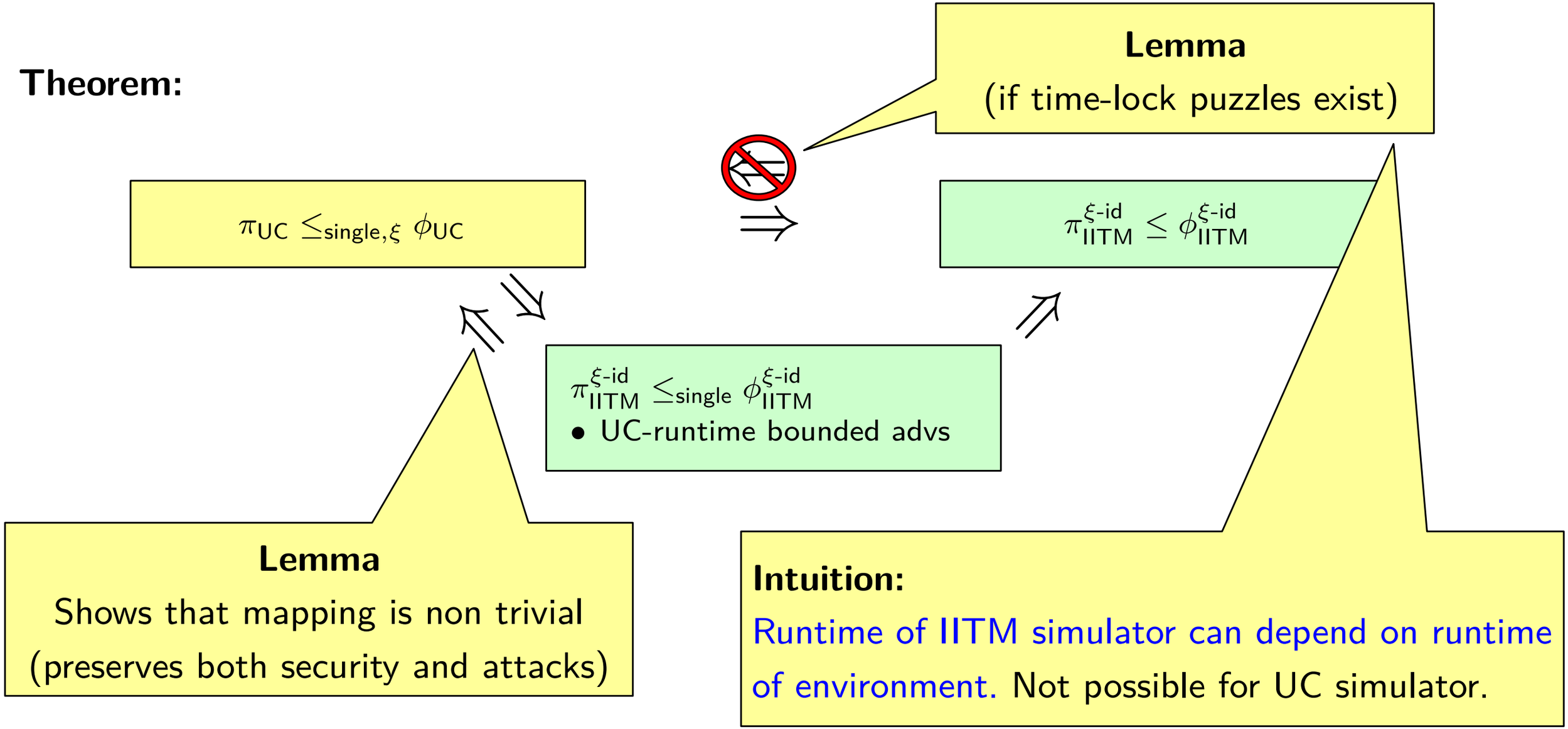
Preserving Security

Theorem:



Preserving Security

Theorem:

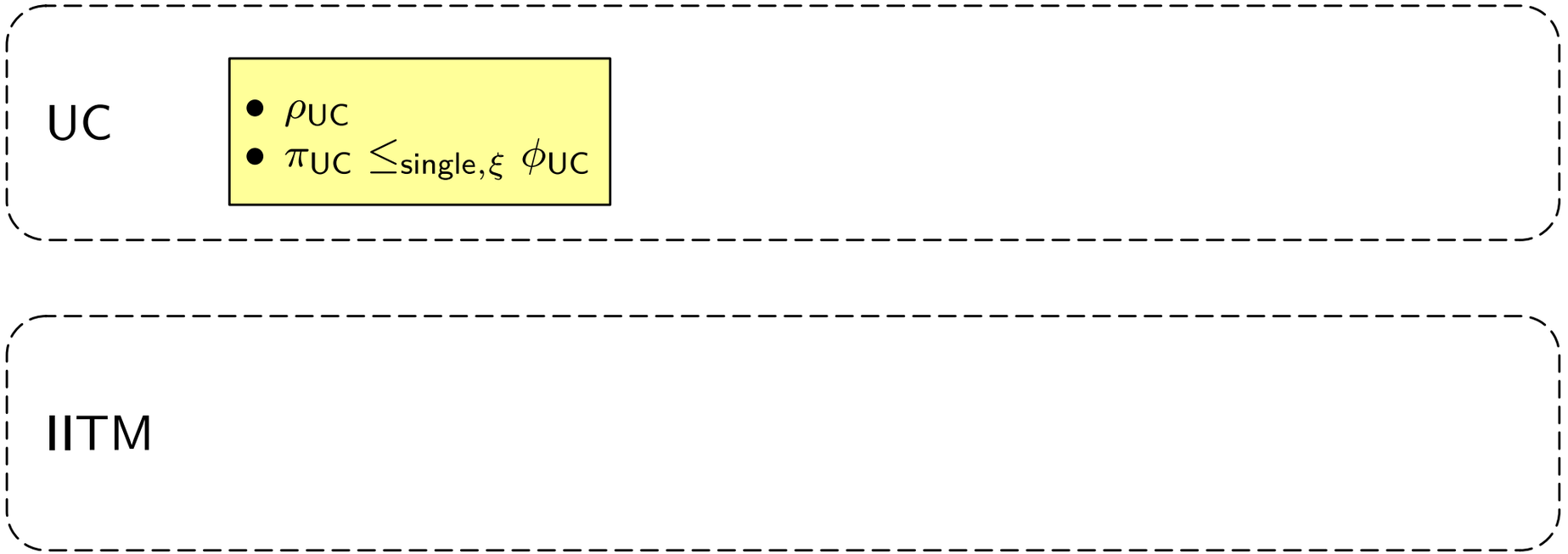


Relating Composition

Corollary: UC composition results also carry over

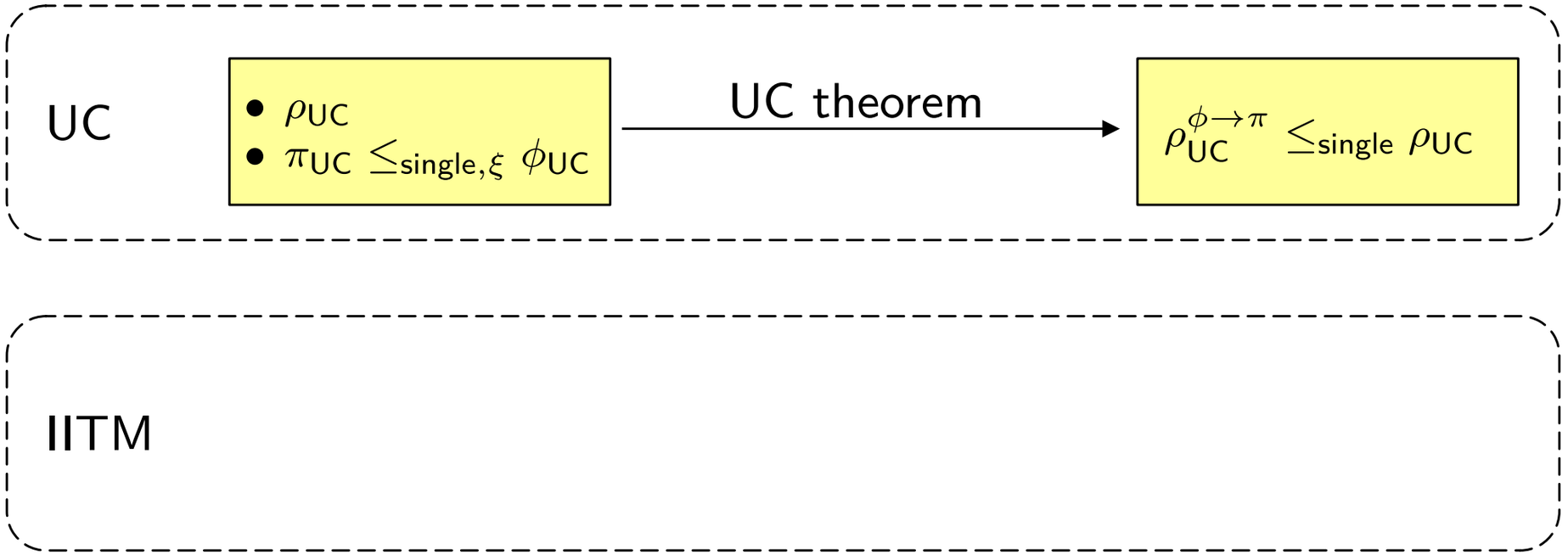
Relating Composition

Corollary: UC composition results also carry over



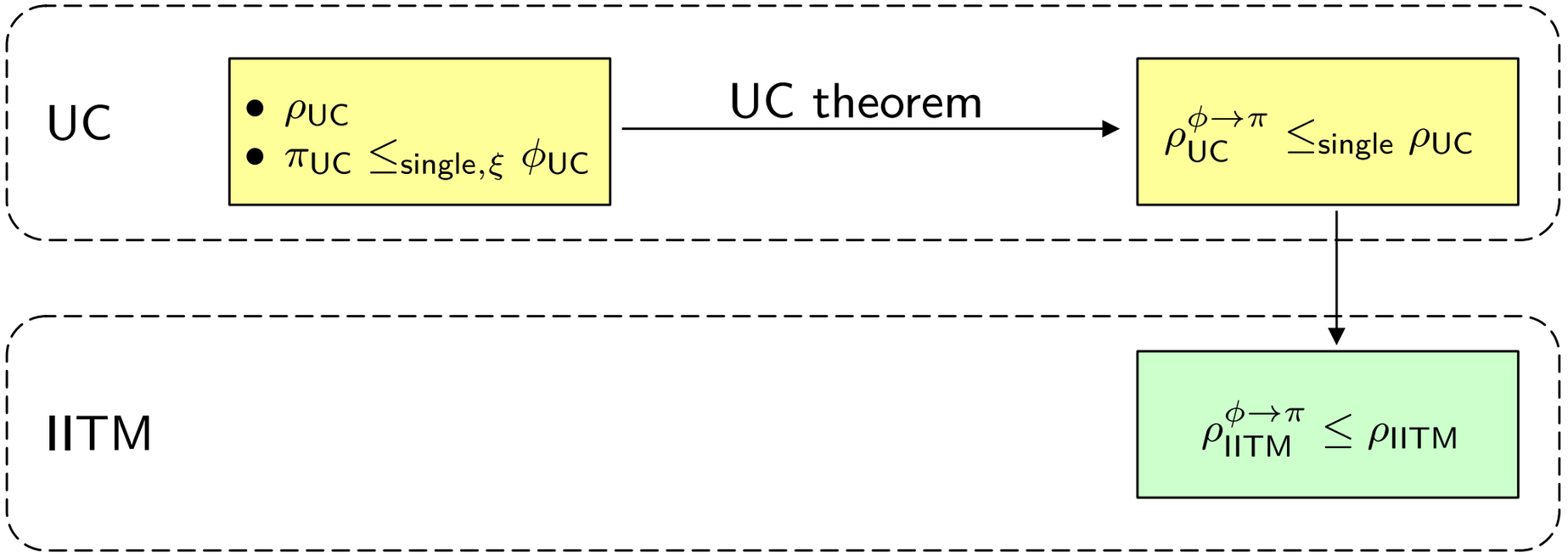
Relating Composition

Corollary: UC composition results also carry over



Relating Composition

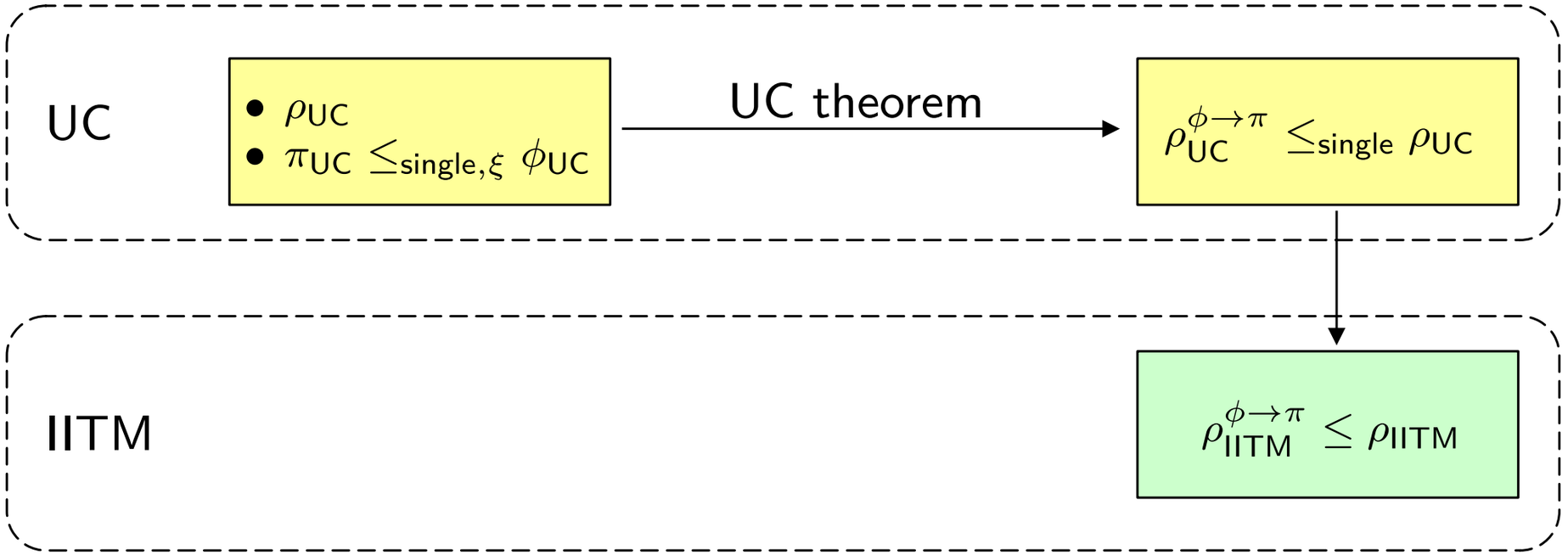
Corollary: UC composition results also carry over



Relating Composition

Corollary: UC composition results also carry over

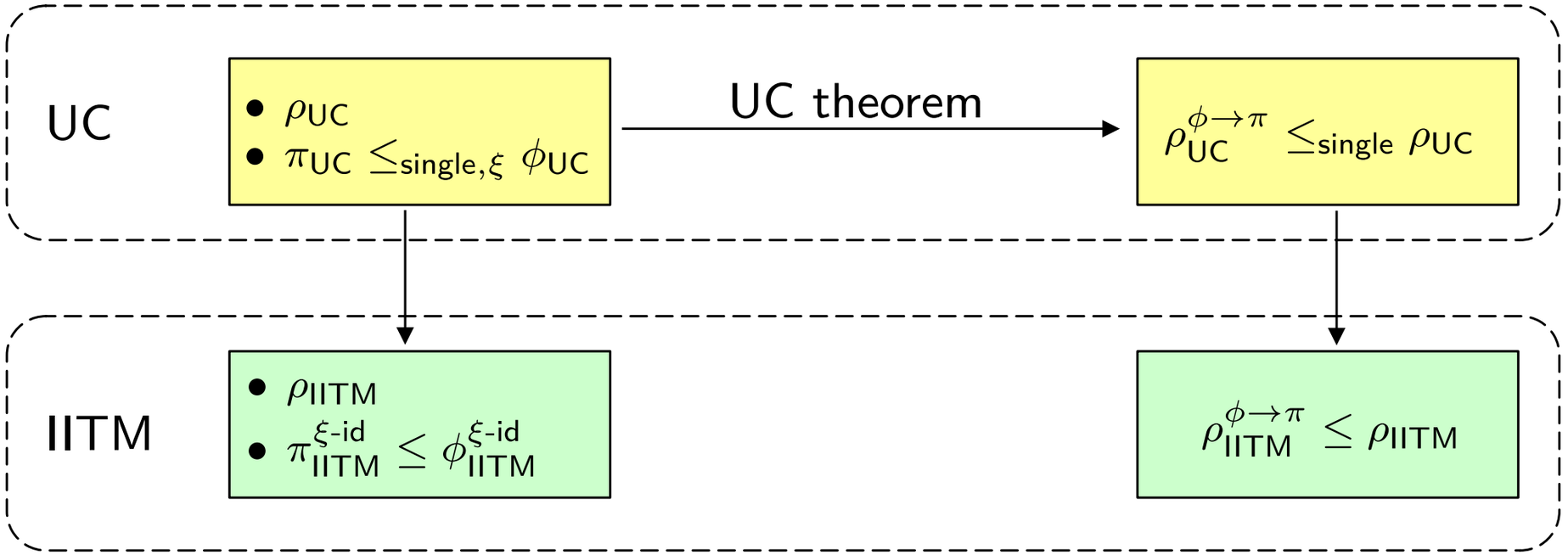
Theorem: Can also directly be obtained in the IITM model



Relating Composition

Corollary: UC composition results also carry over

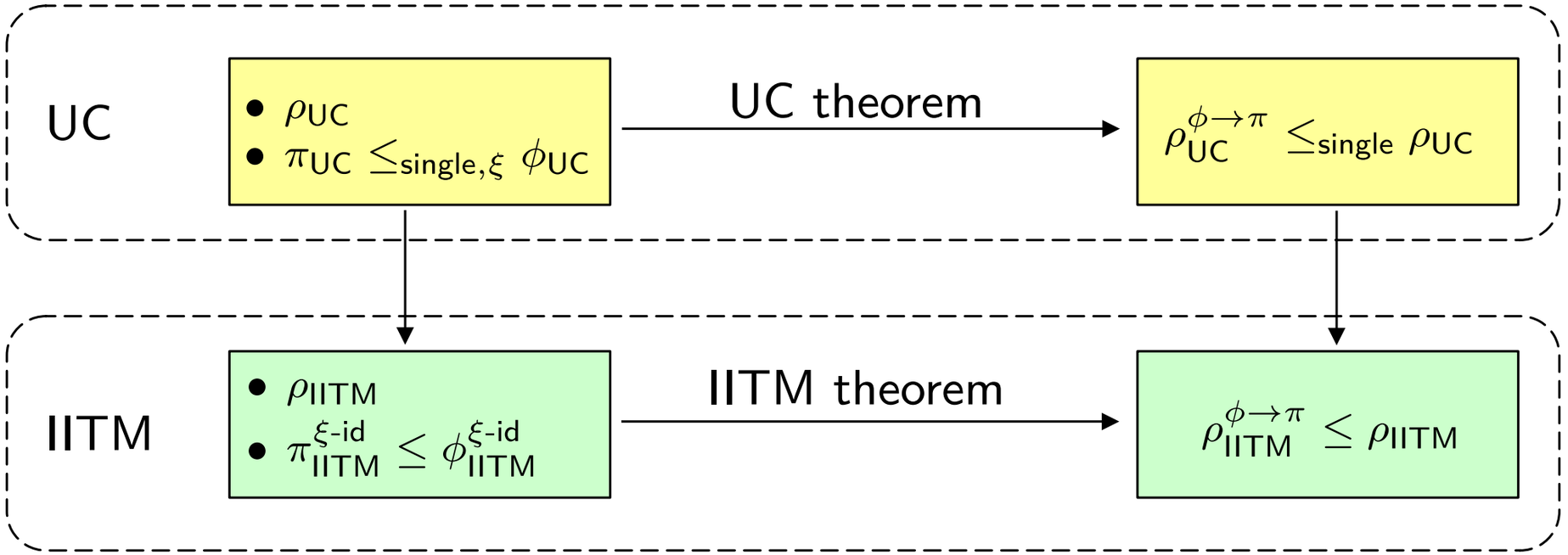
Theorem: Can also directly be obtained in the IITM model



Relating Composition

Corollary: UC composition results also carry over

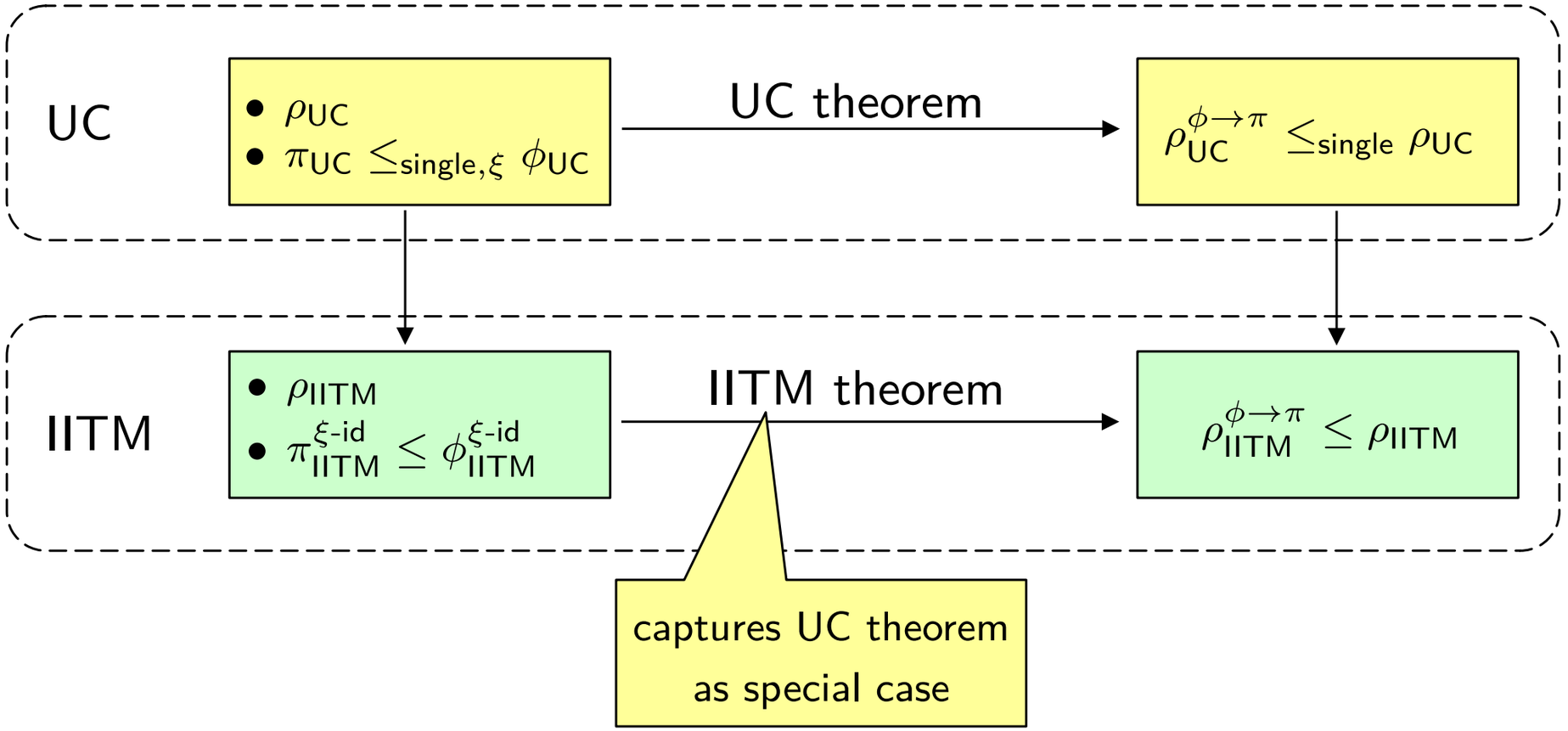
Theorem: Can also directly be obtained in the IITM model



Relating Composition

Corollary: UC composition results also carry over

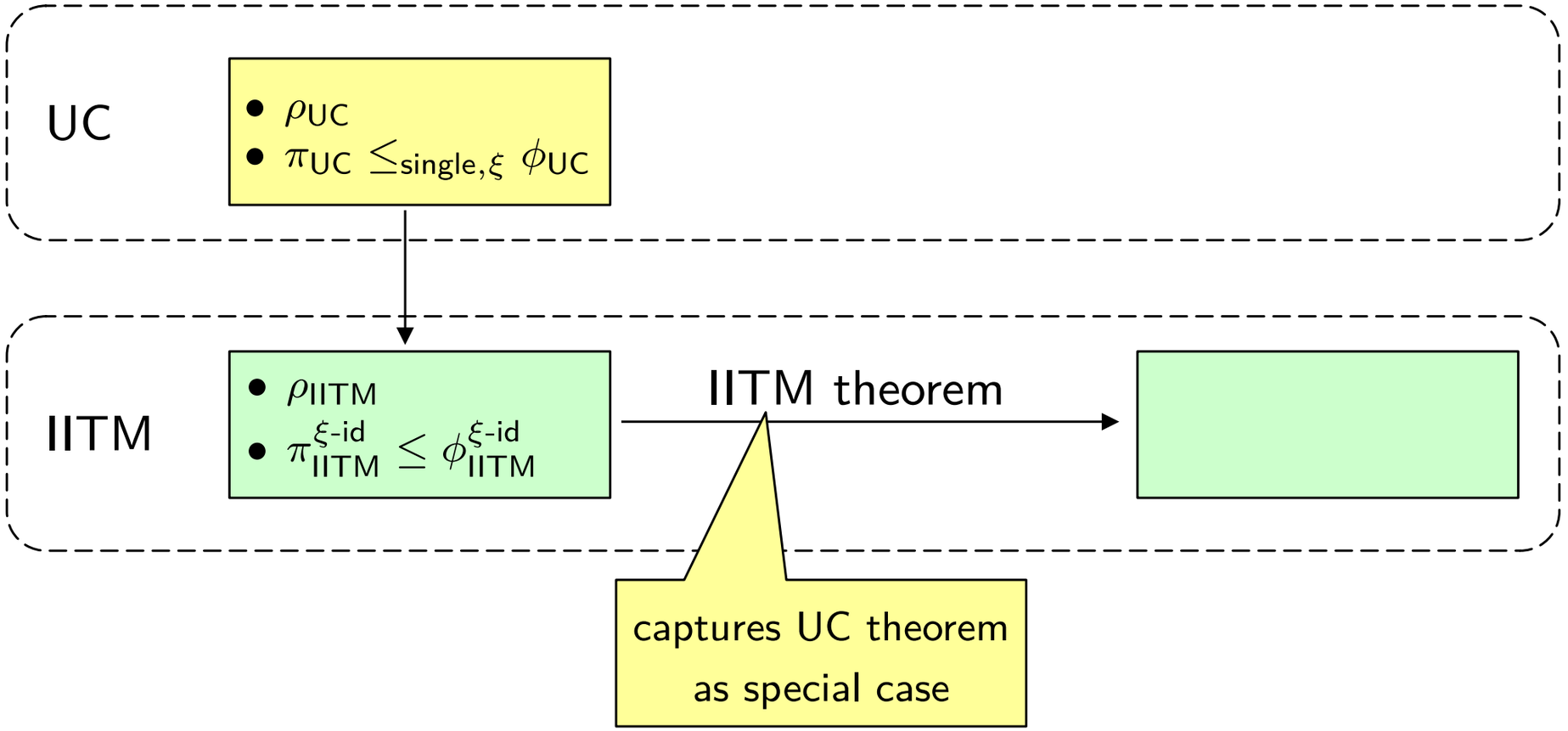
Theorem: Can also directly be obtained in the IITM model



Relating Composition

Corollary: UC composition results also carry over

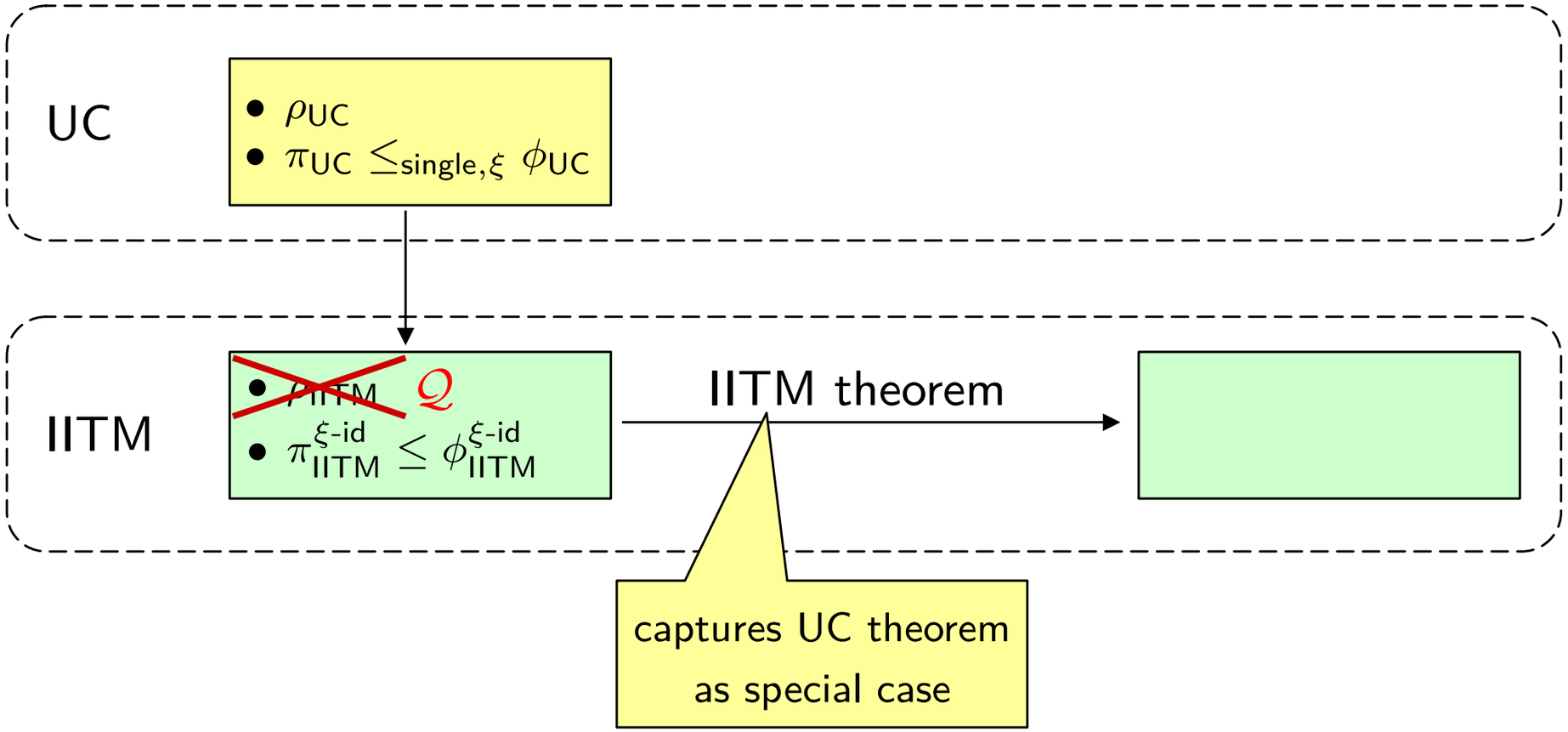
Theorem: Can also directly be obtained in the IITM model



Relating Composition

Corollary: UC composition results also carry over

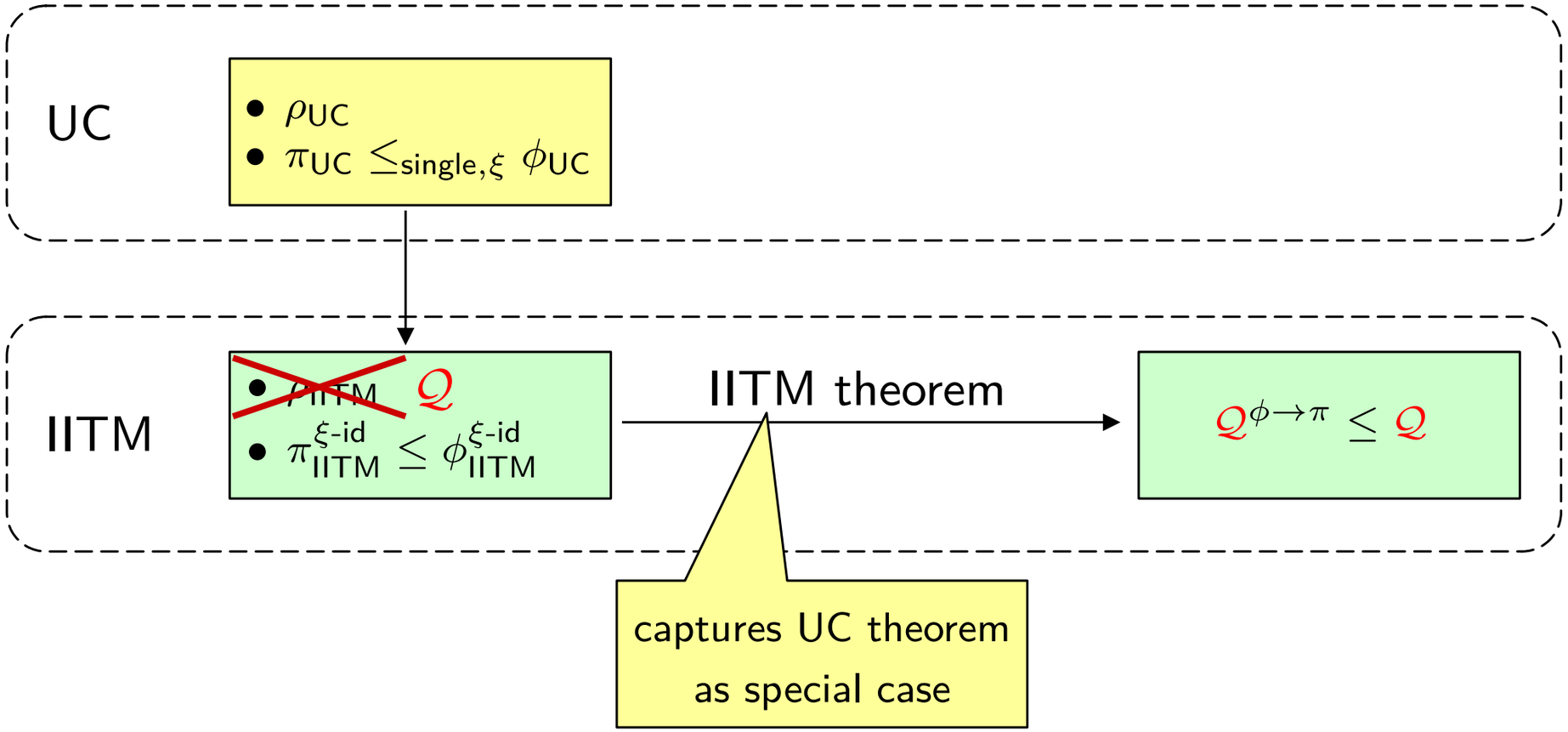
Theorem: Can also directly be obtained in the IITM model



Relating Composition

Corollary: UC composition results also carry over

Theorem: Can also directly be obtained in the IITM model

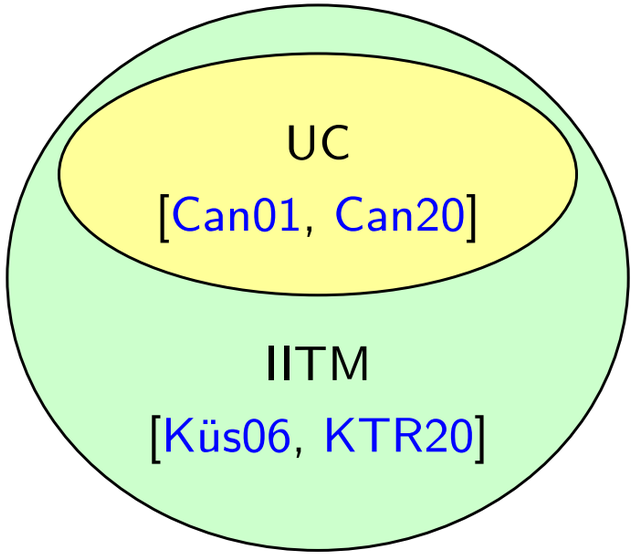


Main Contributions

- Concepts of UC and IITM
- Embedding UC into IITM
- **The Other Direction**

From IITM to UC

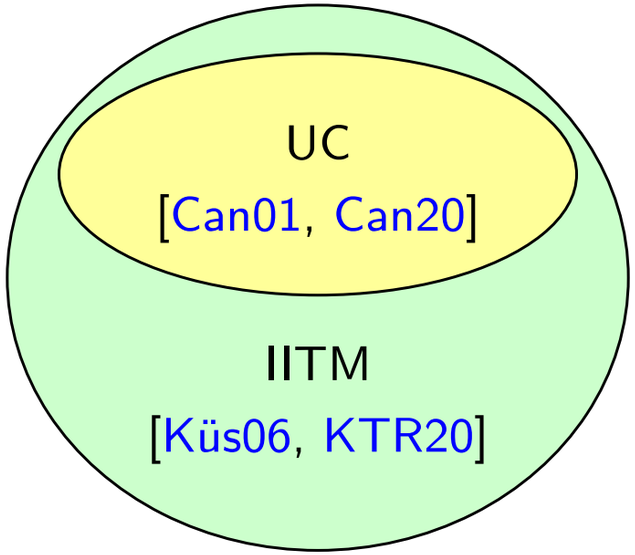
The other direction is impossible in general:



From IITM to UC

The other direction is impossible in general:

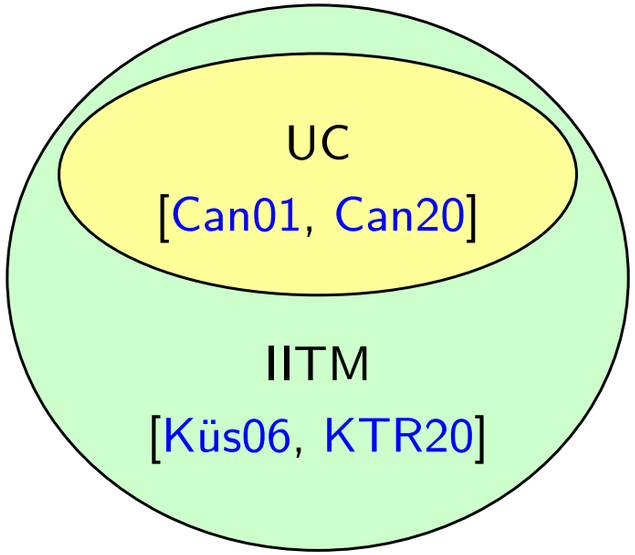
- [KTR20]: There are natural protocols that meet the IITM but not UC runtime notion



From IITM to UC

The other direction is impossible in general:

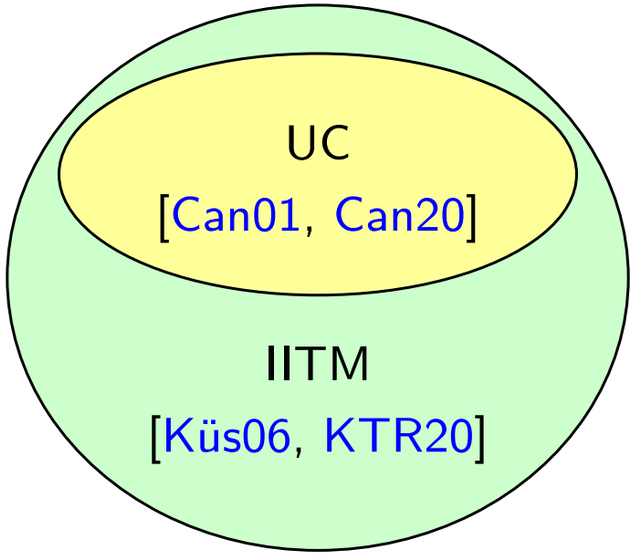
- [KTR20]: There are natural protocols that meet the IITM but not UC runtime notion
- Impossibility result due to simulator classes



From IITM to UC

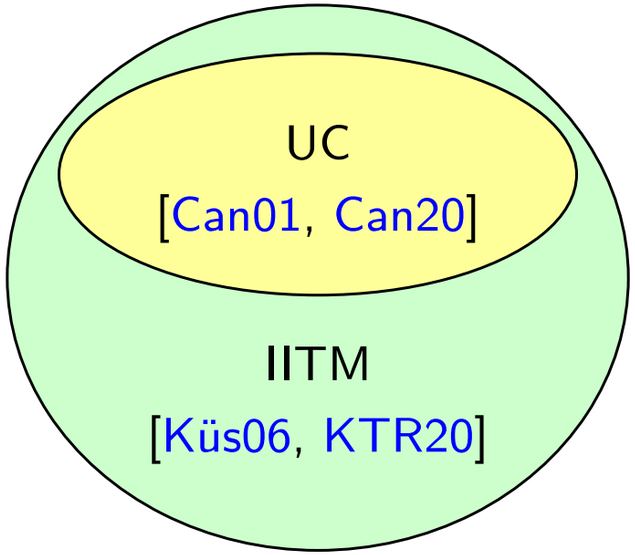
The other direction is impossible in general:

- [KTR20]: There are natural protocols that meet the IITM but not UC runtime notion
- Impossibility result due to simulator classes
- UC protocols must provide an oracle that reveals existence of instances to adversary
  - Changes security properties



Conclusion

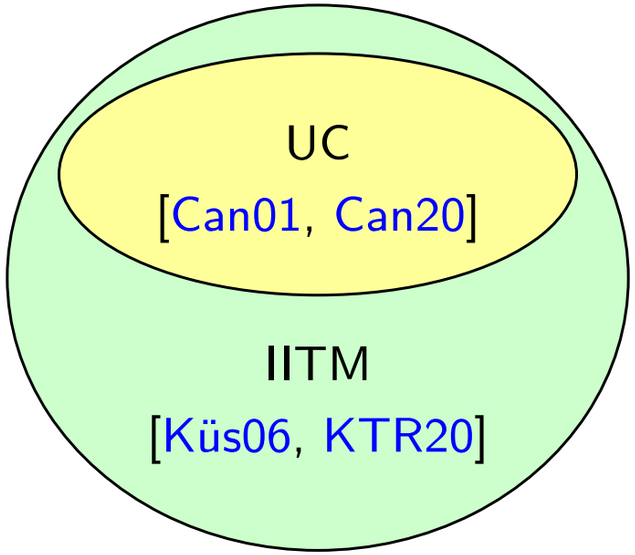
Our work clarifies relationship UC - IITM:



Conclusion

Our work clarifies relationship UC - IITM:

- All UC protocols and results carry over to the IITM Model



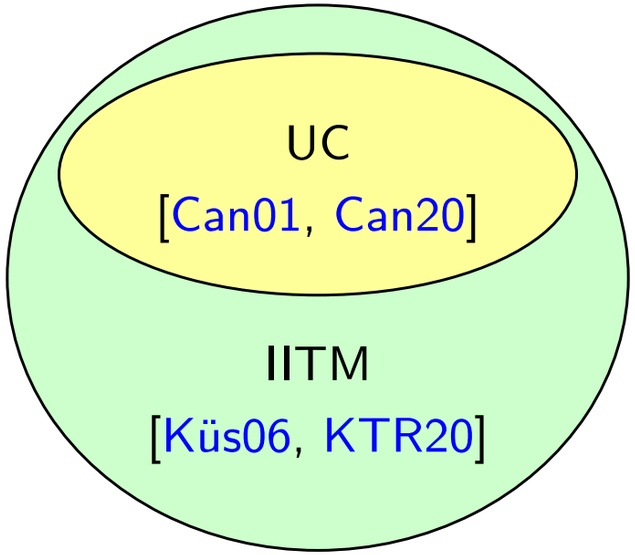
Conclusion

Our work clarifies relationship UC - IITM:

- All UC protocols and results carry over to the IITM Model

  Existing UC results can benefit from IITM features such as

  - joint, global, arbitrarily shared state
  - locally managed SIDs
  - larger classes of simulators and protocols
  - combinations of the above



Conclusion

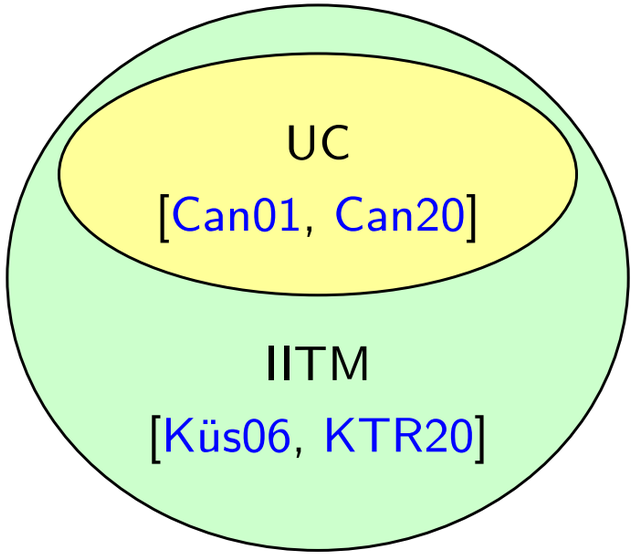
Our work clarifies relationship UC - IITM:

- All UC protocols and results carry over to the IITM Model

  Existing UC results can benefit from IITM features such as

  - joint, global, arbitrarily shared state
  - locally managed SIDs
  - larger classes of simulators and protocols
  - combinations of the above

- Established impossibility results for the other direction



Conclusion

Our work clarifies relationship UC - IITM:

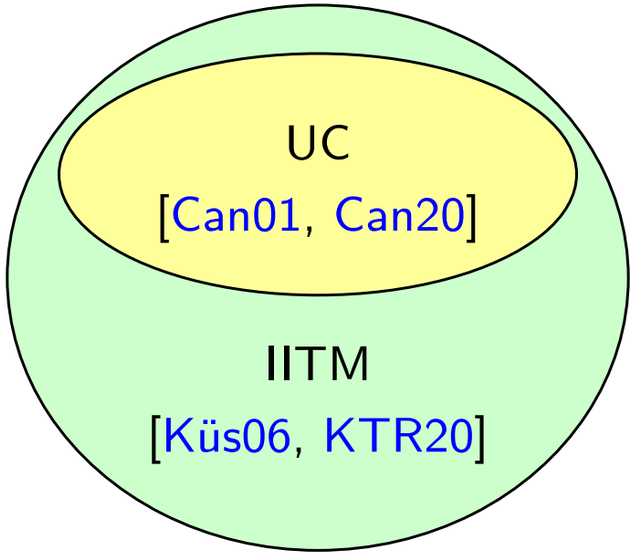
- All UC protocols and results carry over to the IITM Model

  Existing UC results can benefit from IITM features such as

  - joint, global, arbitrarily shared state
  - locally managed SIDs
  - larger classes of simulators and protocols
  - combinations of the above

- Established impossibility results for the other direction
- Future work:

  Identify and map subset of IITM protocols/results to UC



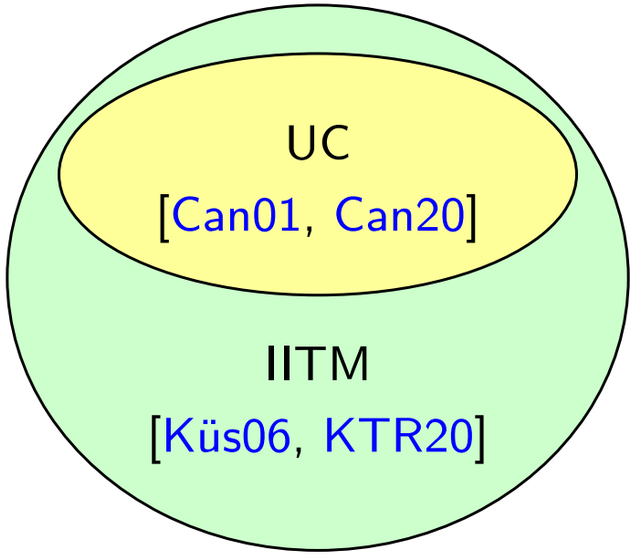
Conclusion

Our work clarifies relationship UC - IITM:

- All UC protocols and results carry over to the IITM Model

  Existing UC results can benefit from IITM features such as

  - joint, global, arbitrarily shared state
  - locally managed SIDs
  - larger classes of simulators and protocols
  - combinations of the above



- Established impossibility results for the other direction
- Future work:

  Identify and map subset of IITM protocols/results to UC

Thanks!