INVERSED TECH

geometry RESEARCH

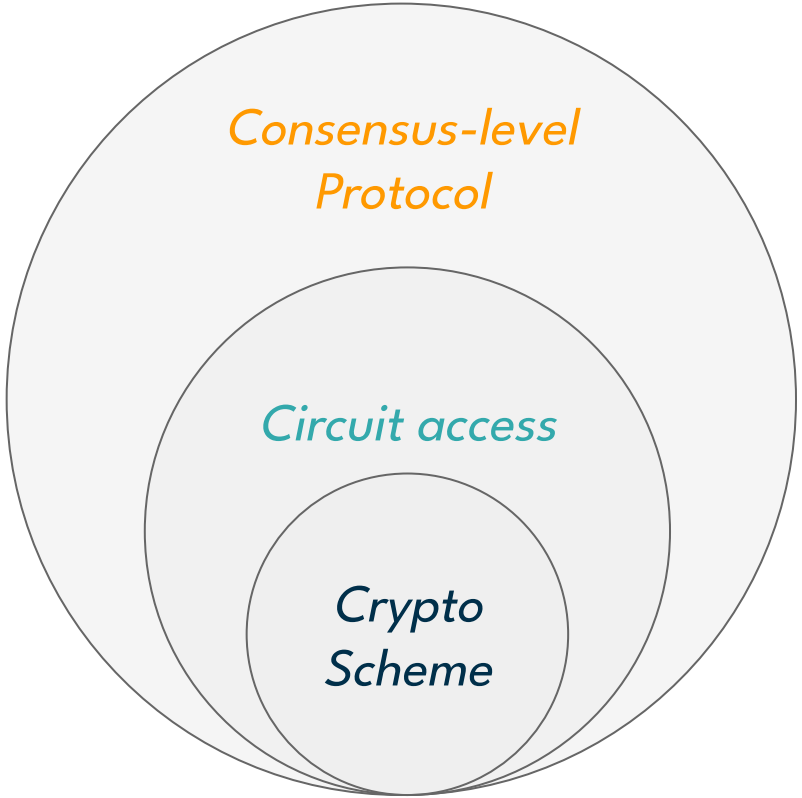
SoK: Programmable Privacy

Daniel Benarroch, Inversed Tech
Ying Tong Lai, Geometry Research

Additional collaborators:
Bryan Gillespie, Inversed Tech
Andrew Miller, UIUC

ZKProof6 Berlin | 2024

The programmability stack



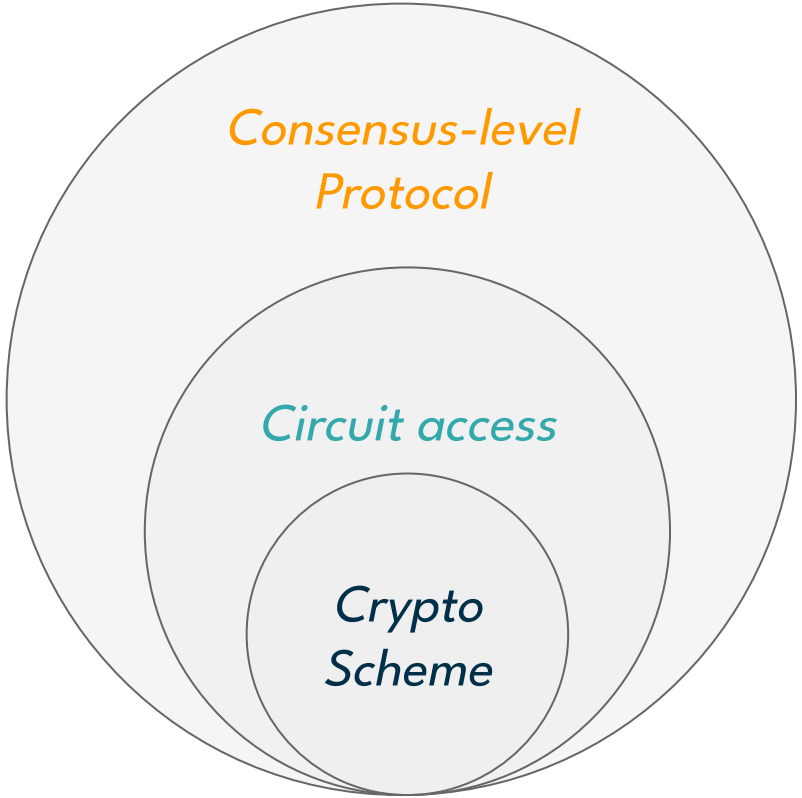
The programmability stack

e.g. Penumbra

aggregation
+ batch decryption

frequent batch auctions

homomorphic threshold
encryption



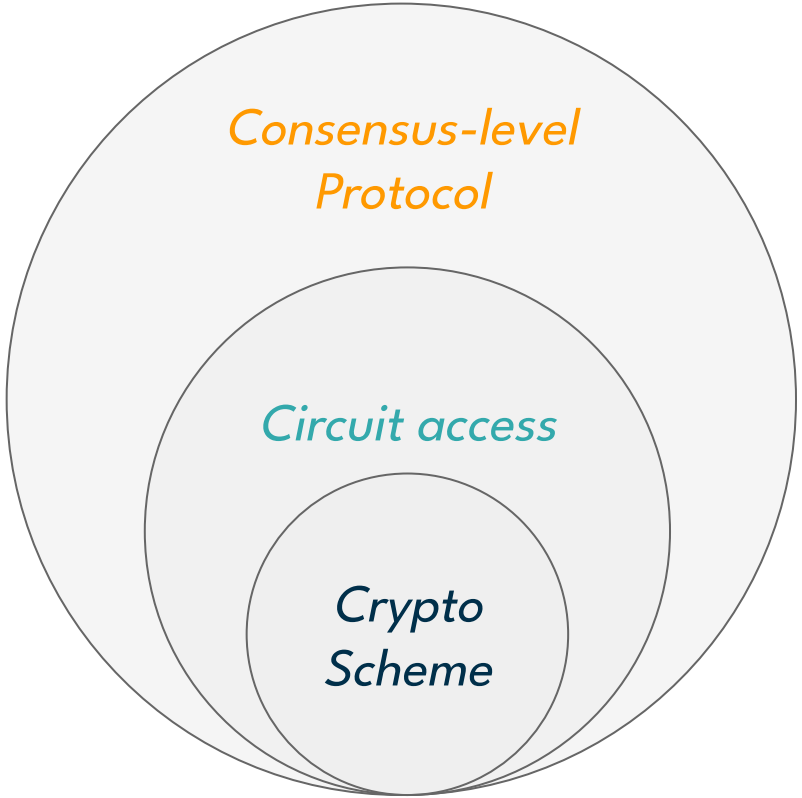
The programmability stack

e.g. Penumbra

aggregation
+ batch decryption

frequent batch auctions

homomorphic threshold
encryption



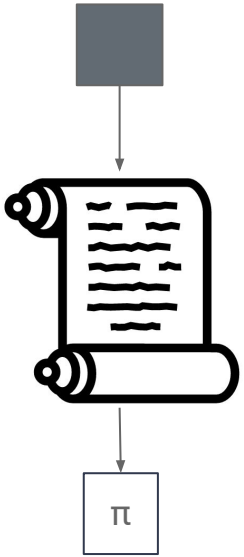
e.g. renegade.fi

proof verification
+ update balances

order-matching +
collaborative proving

two-party computation

Crypto Schemes



programmable disclosure / verification

programmable computation over private data

Consensus-level Protocol

Simple applications

- *individual private state*
- *sequential updates*
- *commutative updates*

(e.g. Zcash, voting, identity)

Interactive applications

- *shared private state*
- *atomic state updates*
- *non-commutative updates*

(e.g. limit order auctions, multilateral trade credit set-off)

Consensus-level Protocol

Simple applications

- *individual private state*
- *sequential updates*
- *commutative updates*

(e.g. Zcash, voting, identity)

Interactive applications

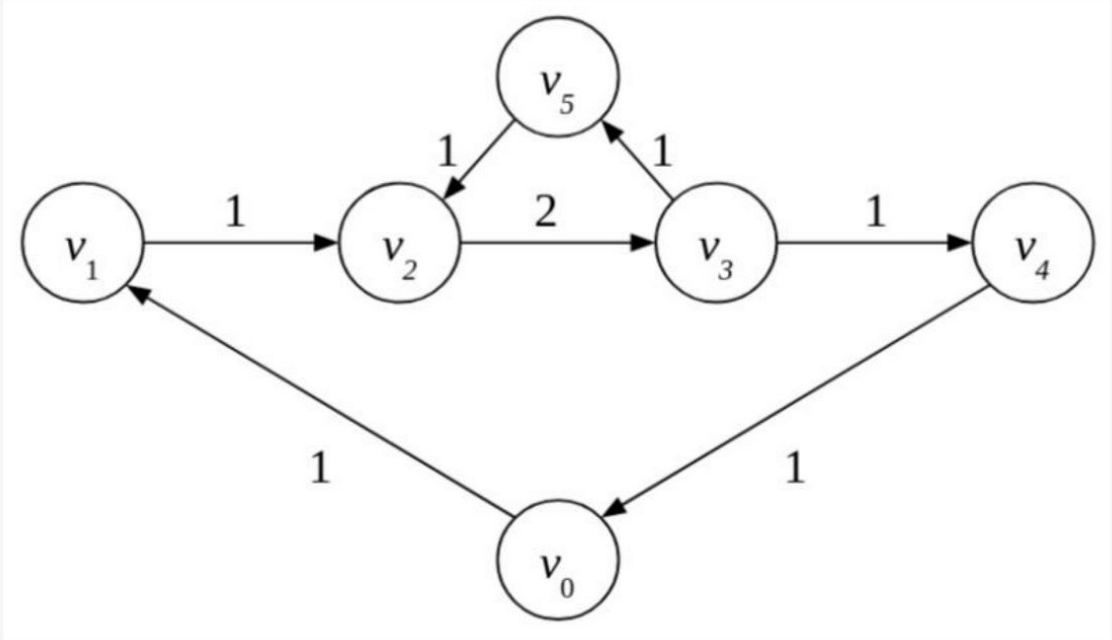
- *shared private state*
- *atomic state updates*
- *non-commutative updates*

(e.g. limit order auctions, multilateral trade credit set-off)

For many interesting applications, ZKP is not enough to provide privacy!

Motivating App: Multilateral trade credit set-off

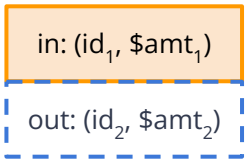
Figure 21. Payment system with a chain and a cycle.



"Liquidity-Saving through Obligation-Clearing and Mutual Credit: An Effective Monetary Innovation for SMEs in Times of Crisis". (2020).

Motivating App: Limit Order Auctions

(order book)

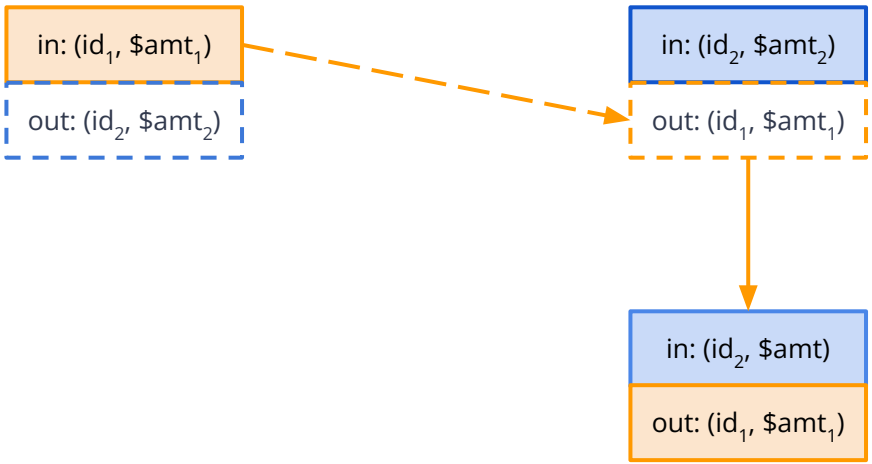


Motivating App: Limit Order Auctions

(order book)

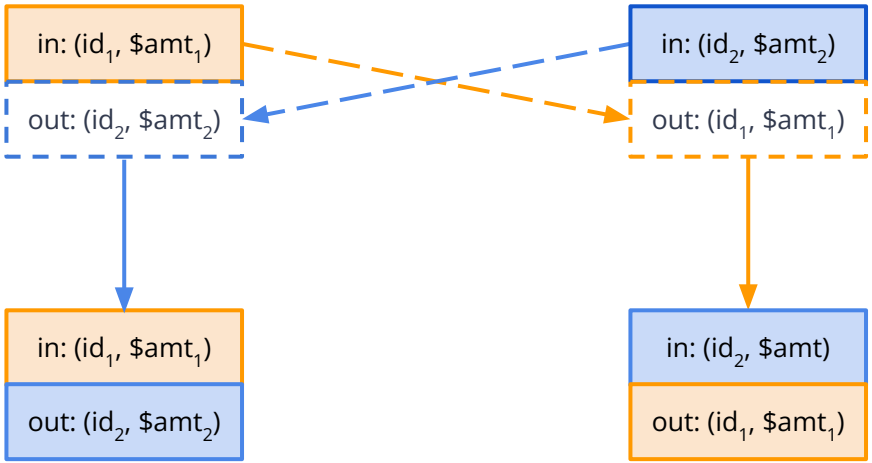
Motivating App: Limit Order Auctions

(order book)



Motivating App: Limit Order Auctions

(order book)

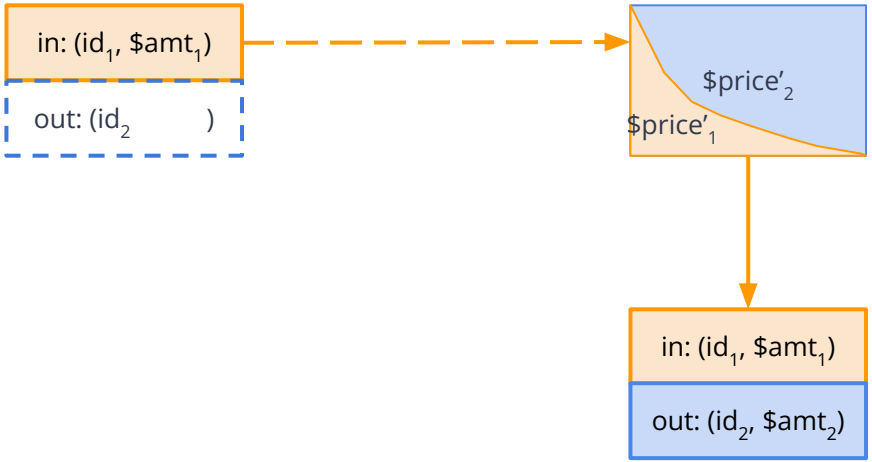


Motivating App: Limit Order Auctions

(CFMM)

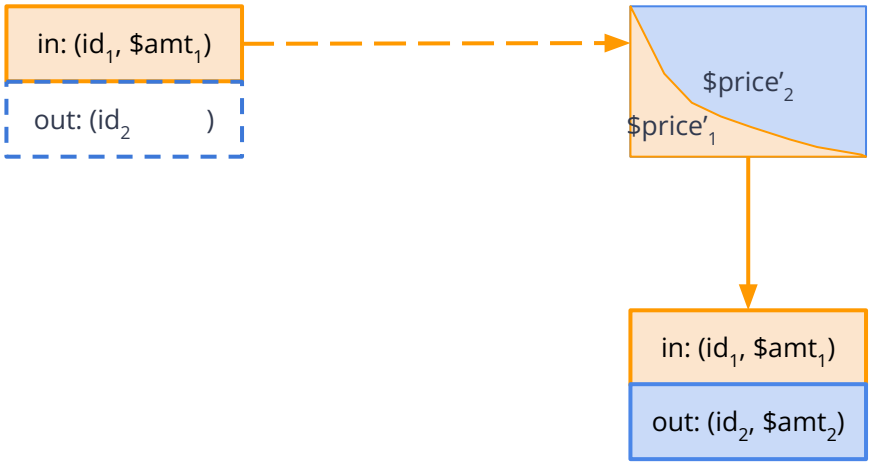
Motivating App: Limit Order Auctions

(CFMM)



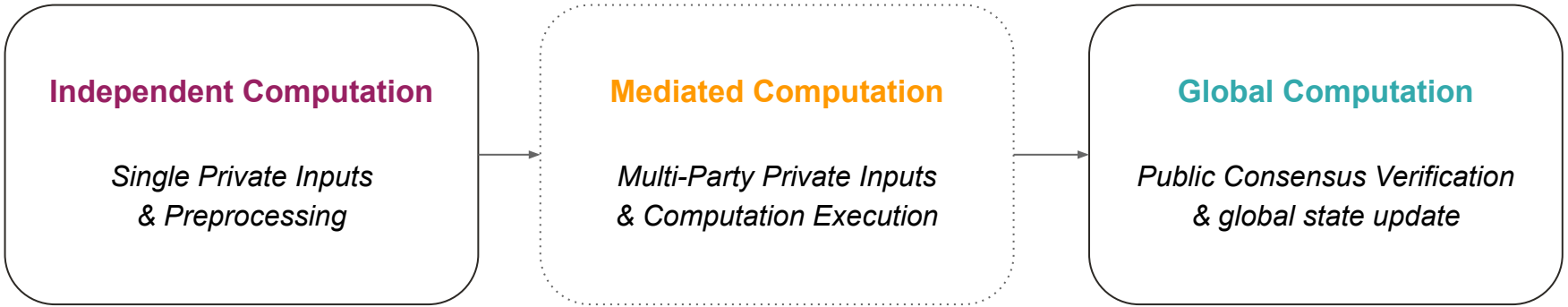
Motivating App: Limit Order Auctions

(CFMM)



price change reveals order;
best we can do is anonymity

Three-Phase Computation Model



Phase 1: Independent computation
On input (Intent, Metadata) from party $\mathcal{P}$:
  add (Intent, Metadata) to IntentSet
  Leak $\mathcal{L}_{\text{Submit}}$(Intent, Metadata, CorruptionState) to $\mathcal{A}$
On input (Intent, Metadata) from $\mathcal{A}$:
  add (Intent, Metadata) to IntentSet

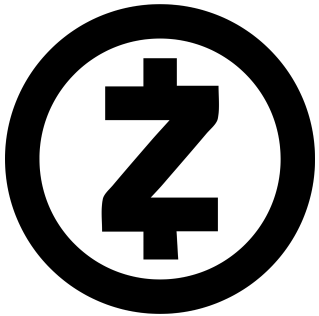
Phase 2: Mediated computation
On input (Advice) from $\mathcal{A}$:
  Result, NewTxs := `MediatedComputatio`
  assert `ComputationAdviceIsValid`(Resul
  add elements of NewTxs to PendingTxs
  If `SatisfiesSecurityAssumptions`(Corr
    Leak $\mathcal{L}_{\text{Med}}^{\text{Expected}}$(Result, IntentSet, AppS
  Else :
    Leak $\mathcal{L}_{\text{Med}}^{\text{Broken}}$(Result, IntentSet, AppSta

Phase 3: Global computation
On input (Advice) from $\mathcal{A}$:
  Result, Txs := `SequenceTransactions`(PendingTxs, AppState, Advice)
  assert `SequencingAdviceIsValid`(Result, CorruptionState)
  For Tx in Txs:
    AppState = UpdateState(AppState, Tx)
    remove Tx from PendingTxs
    append Tx to ExecutedTxs
  Output GlobalView(AppState) to global communication channel
  For each party $\mathcal{P}$:
    Output PrivatePartyView(AppState, $\mathcal{P}$) to $\mathcal{P}$
  Leak PrivateAdversaryView(AppState) to $\mathcal{A}$

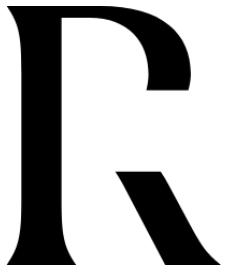
Mediated Computation in Privacy Protocols

Programmability / *Privacy type*	**General-purpose applications**	**Specialized functionality**
k-of-N security	Sunscreen, Zama	Renegade, Penumbra
Hardware security	Secret Network, Obscuro/Ten, Oasis Sapphire, Phala Network, Automata	SUAVE
Extra-protocol	Aztec, Mina, Anoma, AlphaSwap (Aleo), Polygon Miden	ZSA Swap

Example Protocols



ZSA Swap



Renegade.fi

SecretSwap

Penumbra

Example 1: ZSA Swap

in: (id_1, amt_1)

in: (id_2, amt_2)

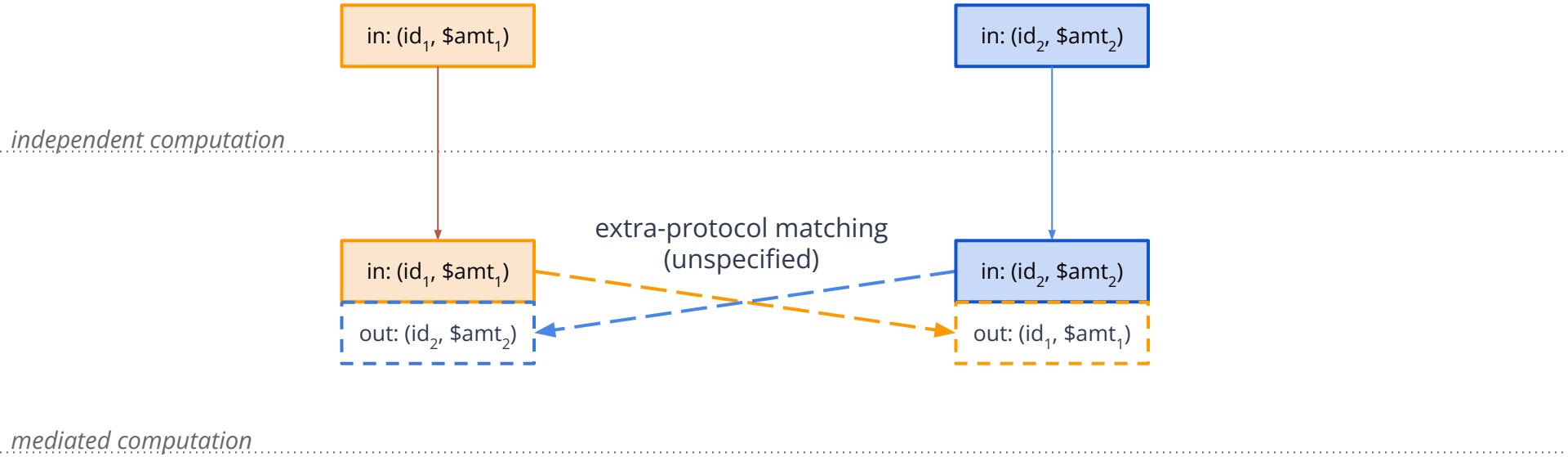
independent computation

mediated computation

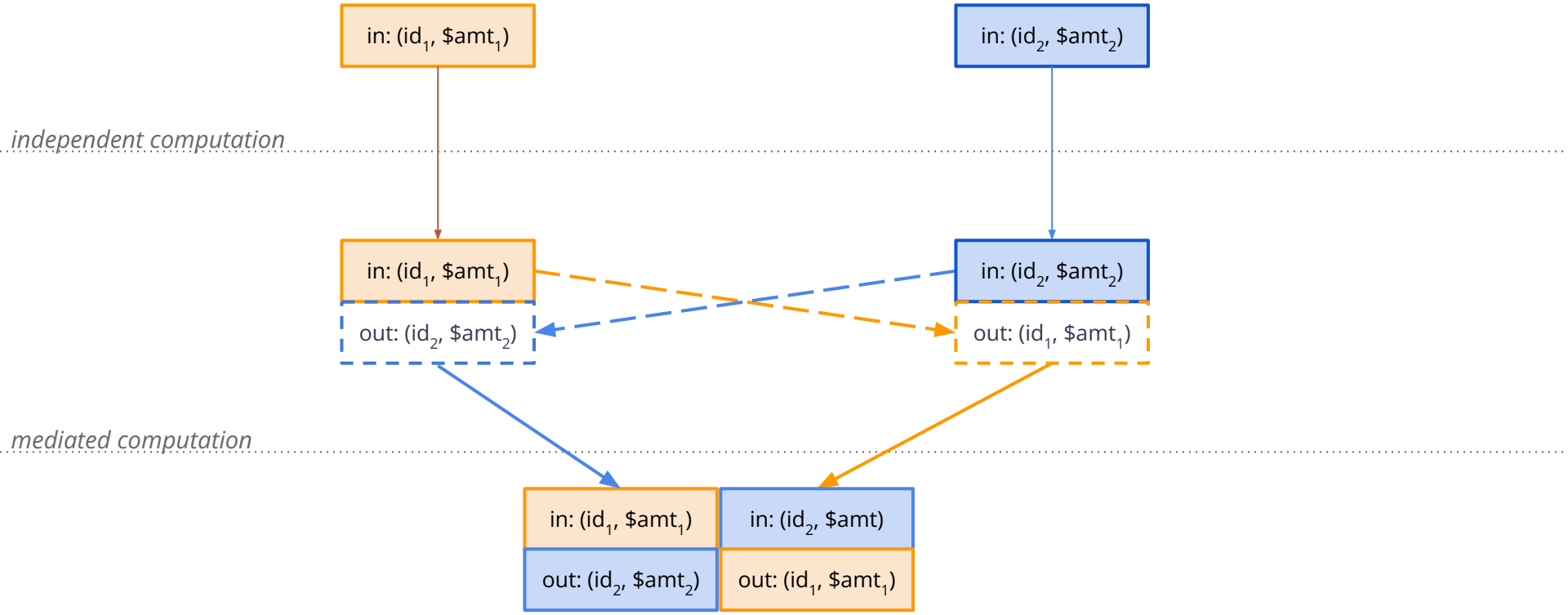
global computation

Example 1: ZSA Swap

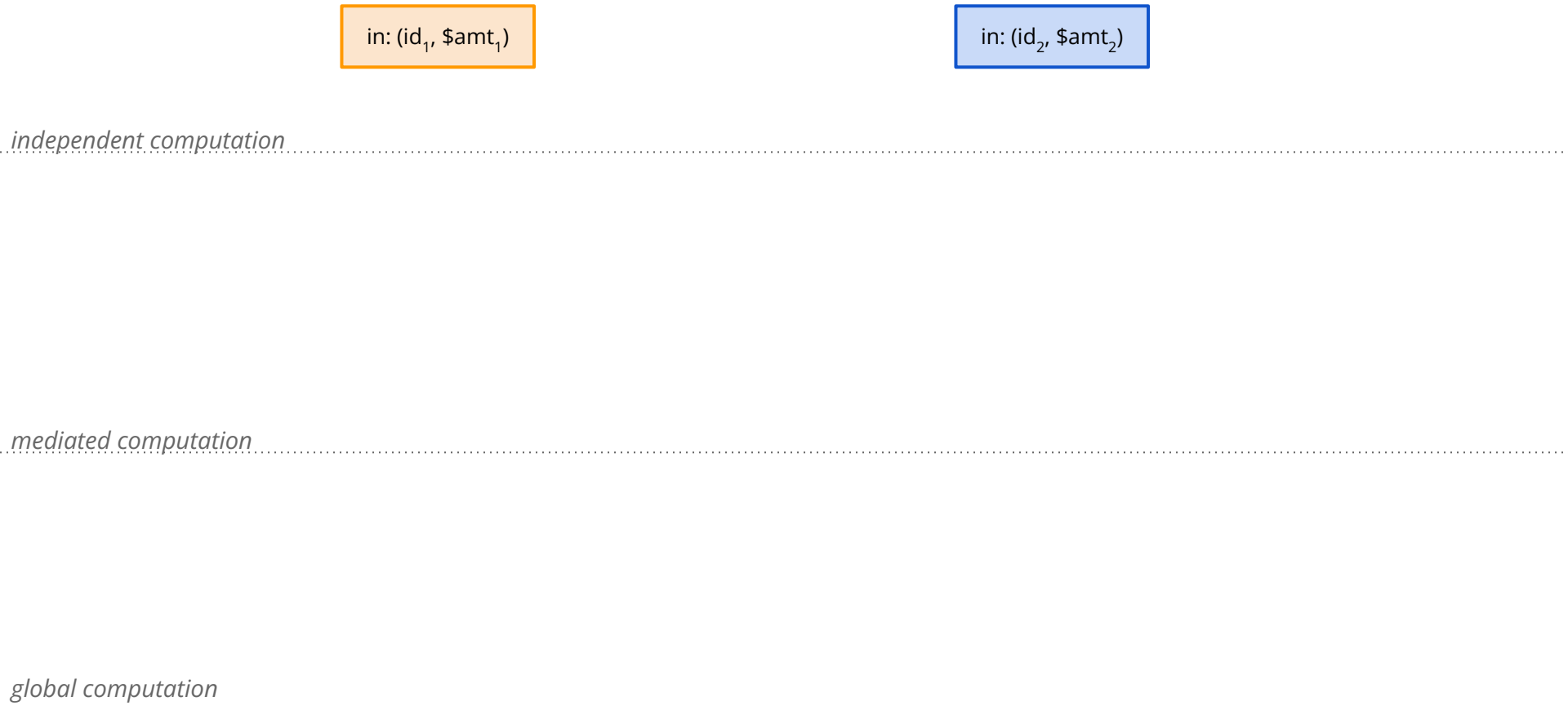
Example 1: ZSA Swap



Example 1: ZSA Swap

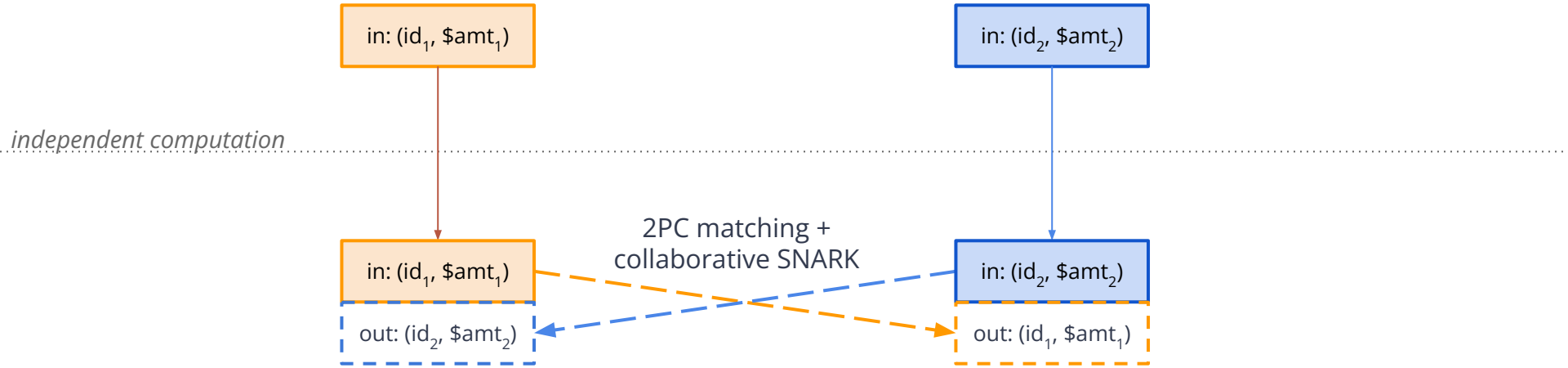


Example 2: Renegade.fi

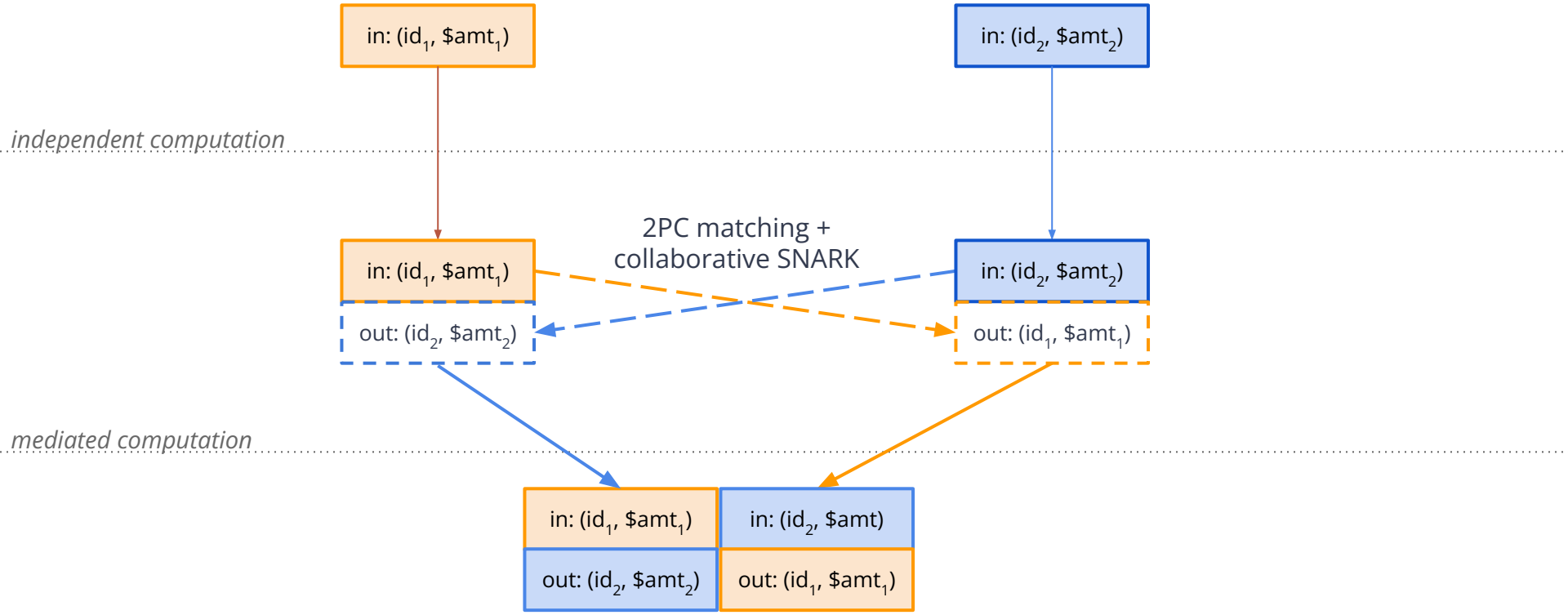


Example 2: Renegade.fi

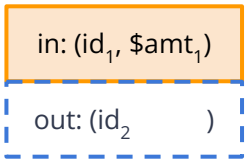
Example 2: Renegade.fi



Example 2: Renegade.fi



Example 3: SecretSwap



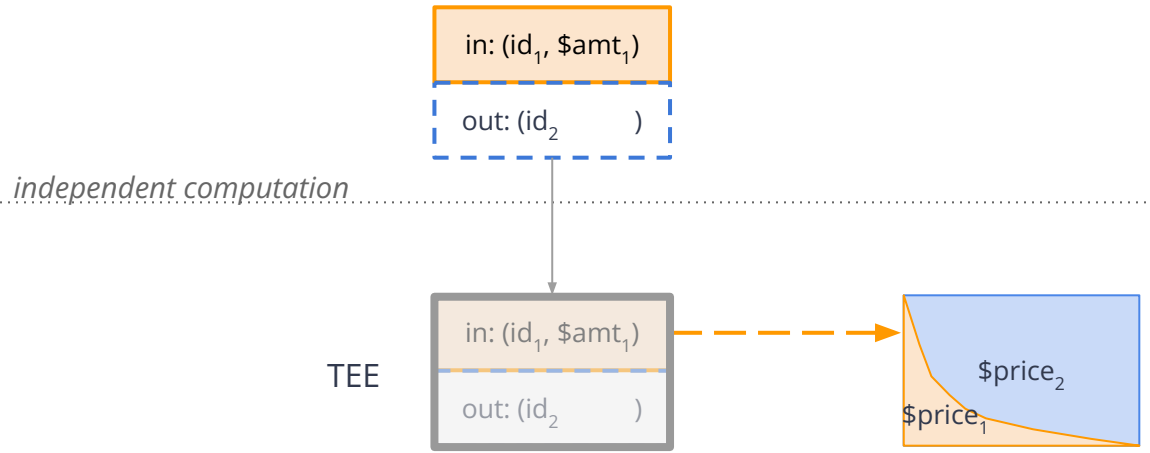
independent computation

mediated computation

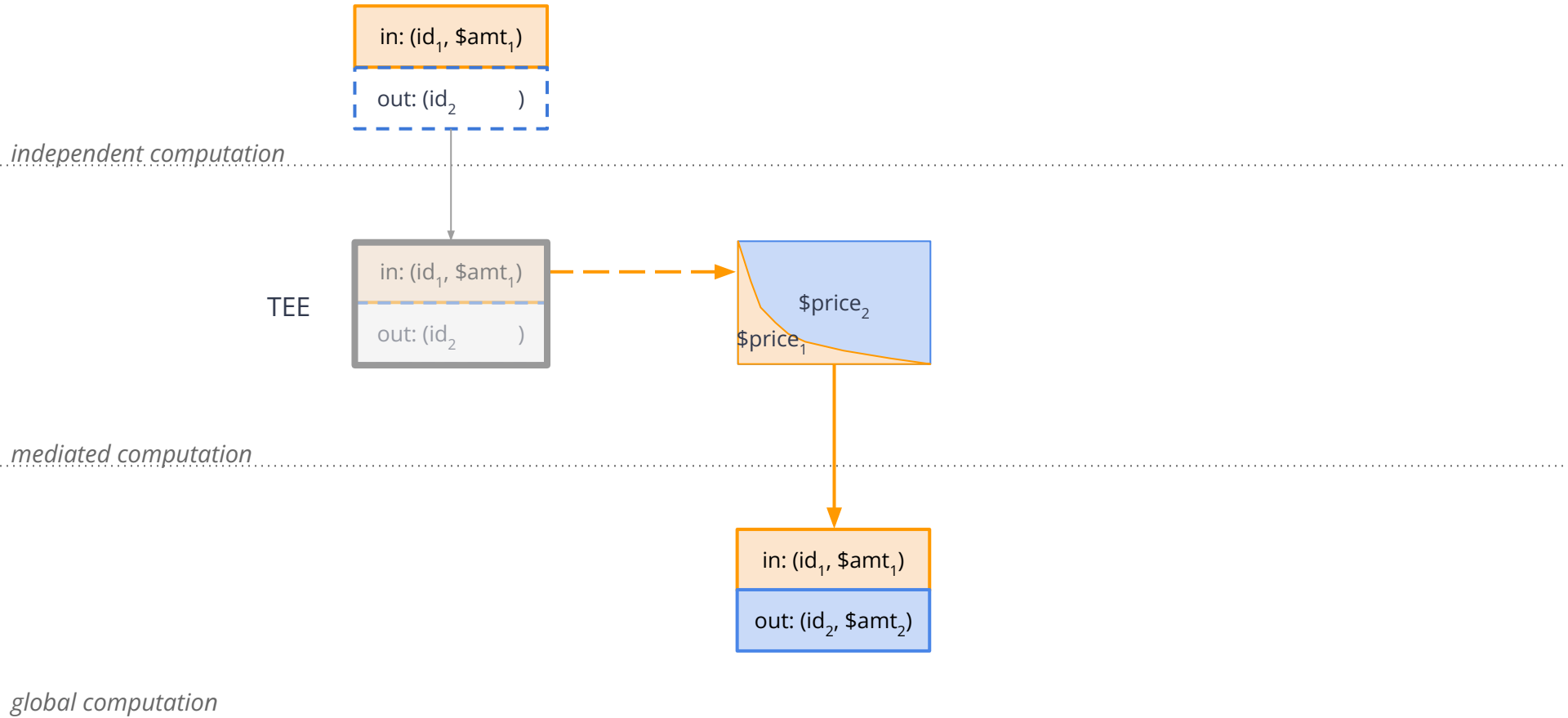
global computation

Example 3: SecretSwap

Example 3: SecretSwap



Example 3: SecretSwap



Example 4: Penumbra

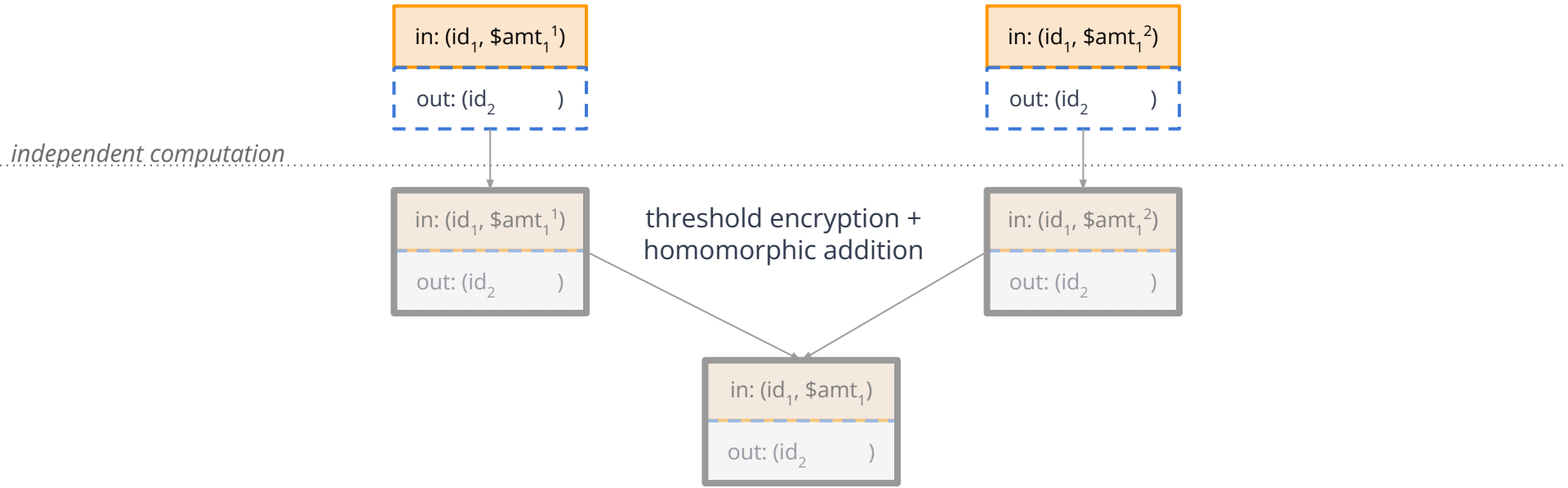
independent computation

mediated computation

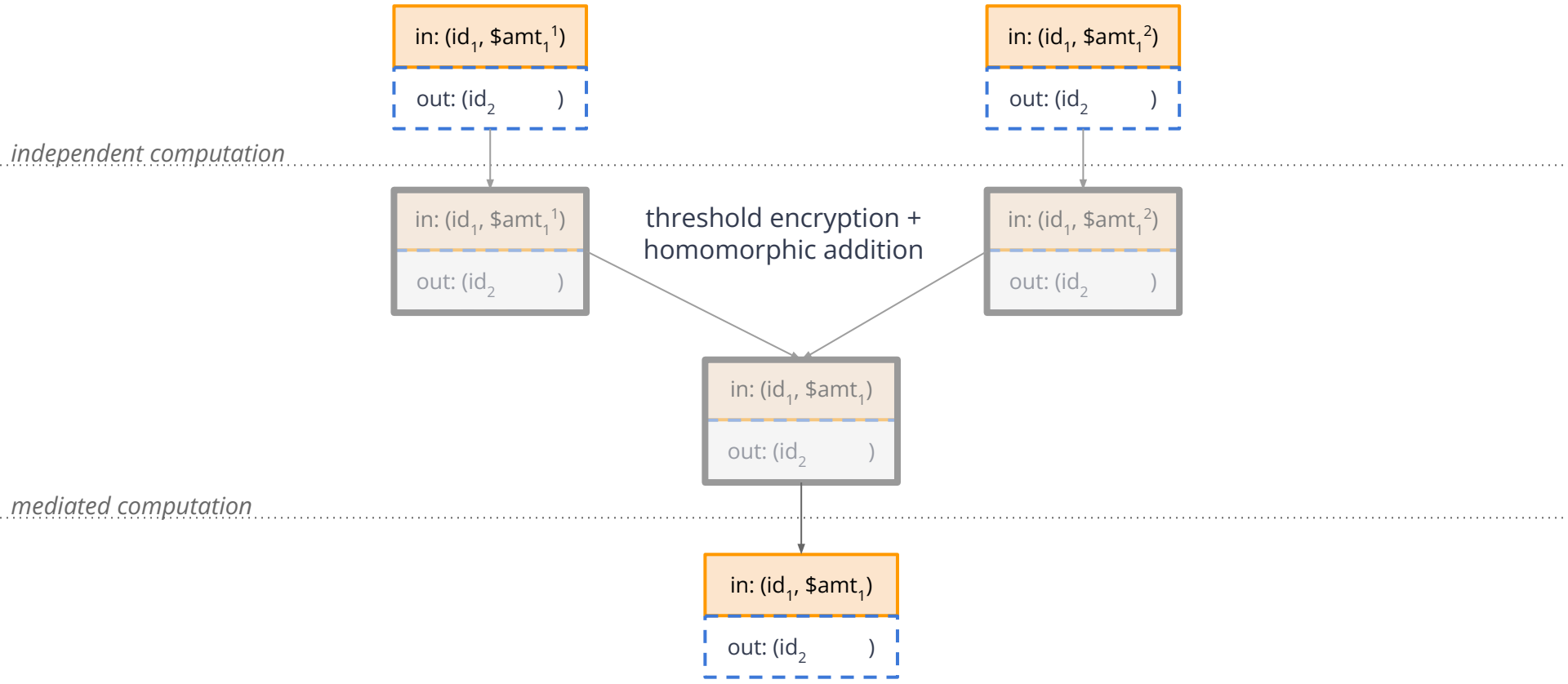
global computation

Example 4: Penumbra

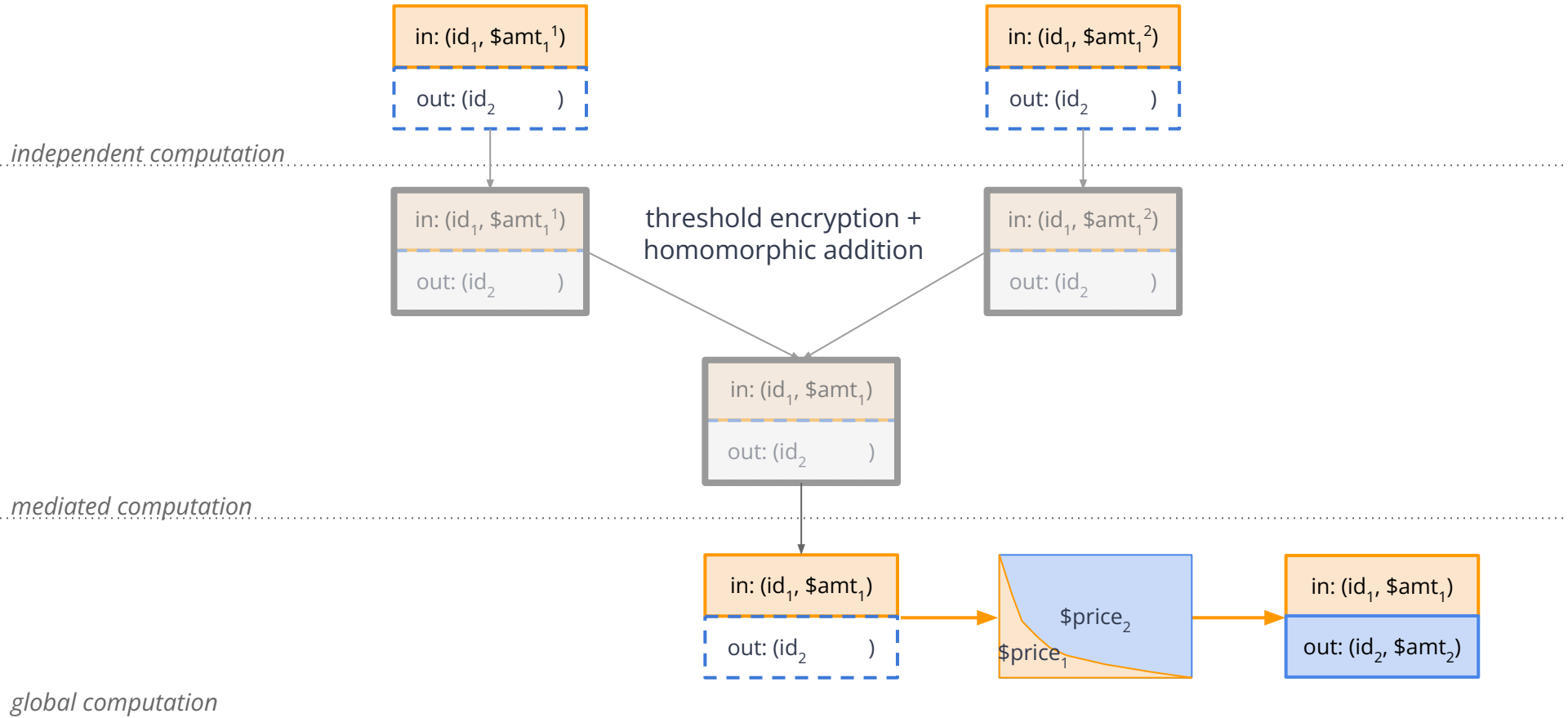
Example 4: Penumbra



Example 4: Penumbra



Example 4: Penumbra



Tradeoff: Security vs. Expressivity

- **In-protocol** mediated computation ⇒ **resource-constrained**
  - e.g. Penumbra, threshold homomorphic encryption
  - e.g. Renegade.fi, 2PC's between relayers

- **Extra-protocol** mediated computation ⇒ **cheaper**, but
  - extra-protocol privacy guarantees fall on application designers
  - potentially fragments state / liquidity across different applications

- Decisions on trade-offs must be **informed by application**
  - e.g. how much order information to reveal is a tradeoff between price efficiency vs. frontrunning prevention
  - tension between enabling **general** applications vs. **optimising** for a single one

Future work / research questions

- **TODO: UC proofs for privacy properties**
  - [done] ideal functionality for three-phase epoch model
  - [done] ideal functionality and protocol descriptions for ZSA Swap, renegade.fi, SecretSwap, Penumbra

- **TODO: Fair metrics**
  - **efficiency**: for a fixed program (e.g. order matching), how much work does a consensus node need to do?
  - **security**: economic cost of attack?

- **TODO: Best practices**
  - defense-in-depth: collaborative proofs; proof of encryption; “remote attestation” zk proof, using TEEs to store key shares in MPC / FHE
  - DSL/compiler safeguards: e.g. prevent writing disclosive MPC circuits

Future work / research questions

TODO: design optimal programmable privacy protocol that fits

1. Fits the three-phase computation model
2. Enables expressiveness for user defined programs while ensuring security of application
3. Provides cryptographic-level privacy

POTENTIAL DIRECTION

→ Include different cryptographic schemes as the “engines” for each phase (e.g.: ZK for phase 1 + FHE or MPC for phase 2)
→ Build a combined DSL that knows to speak to both engines with a fixed separation as soon as inputs need to be aggregated
→ Define privacy **invariants** at the language level to prevent unintentional leakage
→ Provide high-level primitive implementations for users to use off-the–shelf, increasing security

THANK YOU
QUESTIONS?

INVERSED TECH

@inversed_tech

geometry
RESEARCH

@__geometrydev__