Zero Knowledge Proofs

ZKP Applications Overview & zkBridge, Trustless Bridge Made Practical

Instructors: Dan Boneh, Shafi Goldwasser, **Dawn Song**, Justin Thaler, Yupeng Zhang





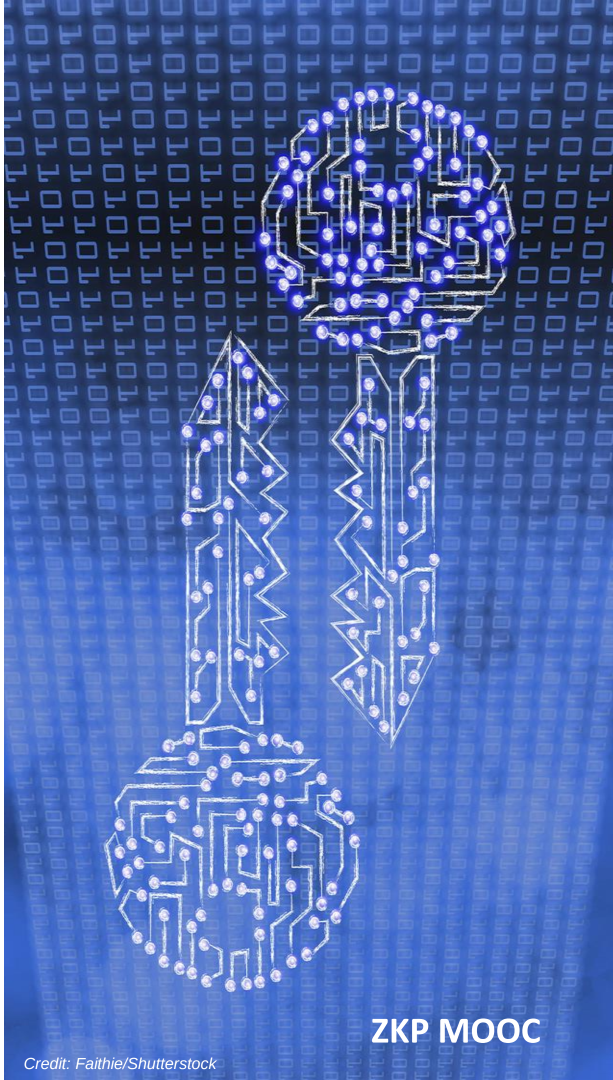
Stanford University

Berkeley UNIVERSITY OF CALIFORNIA

GEORGETOWN UNIVERSITY

TEXAS A&M UNIVERSITY

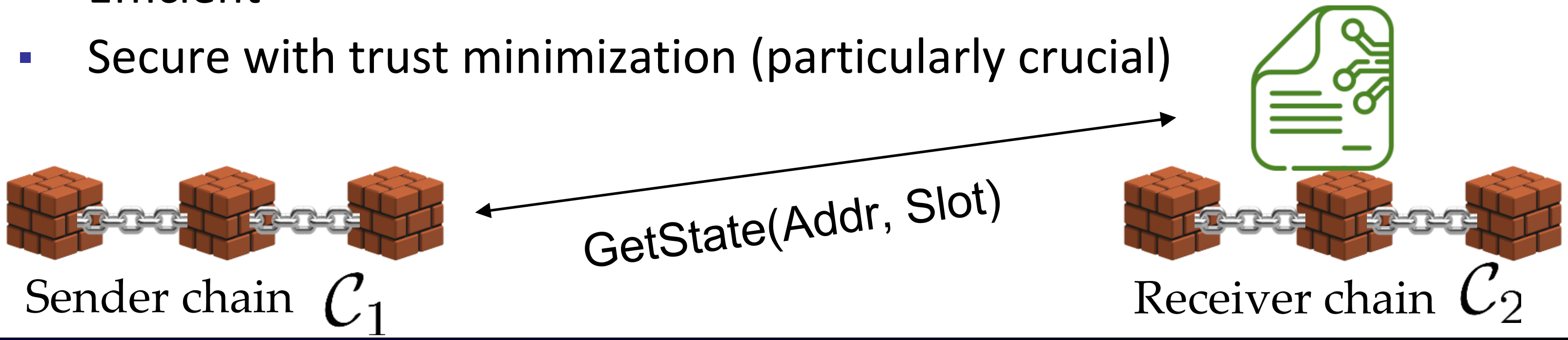
zkBridge: Trustless Bridge Made Practical



Credit: Faithie/Shutterstock

Cross-chain Bridges

- Multi-chain Universe
- Bridge: generic and efficient communication cross blockchains
- Desirable properties
  - Generality (support many applications)
  - Efficient
  - Secure with trust minimization (particularly crucial)



Current Common Bridge Approach: Trust Intermediary

Existing Approach: intermediary

- Side chain (PolyNetwork, Axelar)
- Committee (Wormhole, Ronin)
- Oracles (LayerZero)

Trust Assumptions

- 2/3 honest nodes
- 2/3 honest committee
- independence between Oracle and Relayer

Pros: Simple & efficient on-chain verification (e.g., multisig)
Cons: Need to rely on external trust on intermediaries

Over $2B Lost in Cross-chain Bridge Attacks in last 18 months

Bridge Protocol	Hacked Time	Total Loss
BSC Bridge	2022-10	$568M
Nomad	2022-08	$200M
Harmony	2022-06	$100M
Ronin	2022-03	$600M
Wormhole (Solana)	2022-02	$325M
PolyNetwork	2021-08	$600M

Cause: Private Key Leakage

Remove Trust on Intermediary

- Light client verification:
  - Verifying certain correctness properties of state transition in consensus protocol
  - E.g., for BFT-based consensus, a light client needs to verify validator signatures and keeps track of validator rotation
- Cosmos IBC
  - Validators verifies block header information of another chain, performing light client verification
  - Cons: require each chain to implement IBC client to perform the verification
- NEAR Rainbow bridge
  - Implement light client verification as a smart contract in Ethereum
  - Cons: on-chain verification is very expensive

zkBridge—Trustless Bridge Made Practical

- With ZKP, we replace **honesty assumptions** with **Cryptographic assurance**

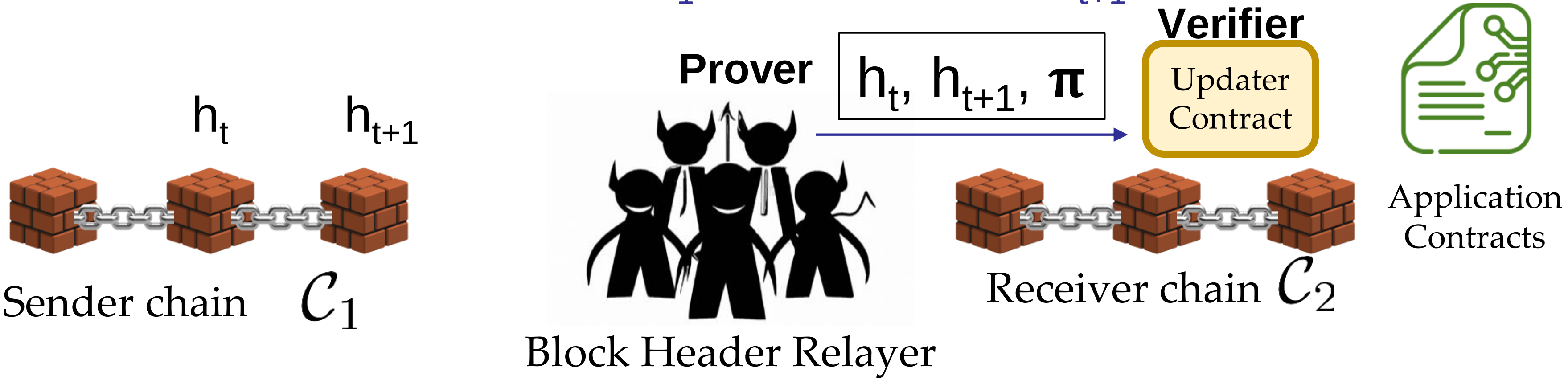
Zero-knowledge proofs

- **Efficient on-chain verification using ZKP**

Xie-Zhang-Cheng-Zhang-Zhang-Jia-Boneh-Song, "zkBridge: trustless bridge made practical", ACM CCS 2022 (zkbridge.org)

zkBridge—Trustless Bridge Made Practical

- **π**: proving h_{t+1} is correct given h_t (and other info) (consensus-specific light client verification) with SNARKs
- E.g., "∃ sigs by a majority of C_1 committee on h_{t+1}"



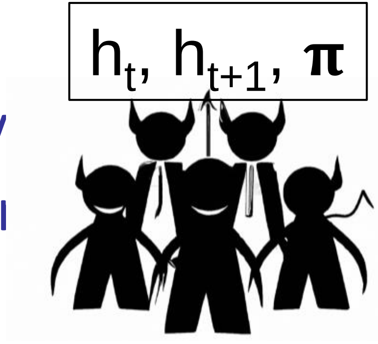
Xie-Zhang-Cheng-Zhang-Zhang-Jia-Boneh-Song, "zkBridge: trustless bridge made practical", ACM CCS 2022 (zkbridge.org)

Advantages of zkBridge (zkbridge.org)

- Minimized trust
  - Cryptographic soundness instead of honest assumptions
- Efficient on-chain verification
  - purpose-built zkSNARK enables efficient on-chain verification
- Permissionless and Decentralized
  - Provers are not trusted so anyone can join
- Extensible and Universal
  - Developers can develop their own application on top

Challenges

- SNARKs are expensive
- Blockchains are not designed to be "ZK friendly"
  - EdDSA digital signature is expensive to express as an arithmetic circuit (~2M gates)
- Each state transition can involve hundreds of sig v
- => Computing **π** naively can be prohibitively expe



Block Header Relayer

Making zkBridge practical

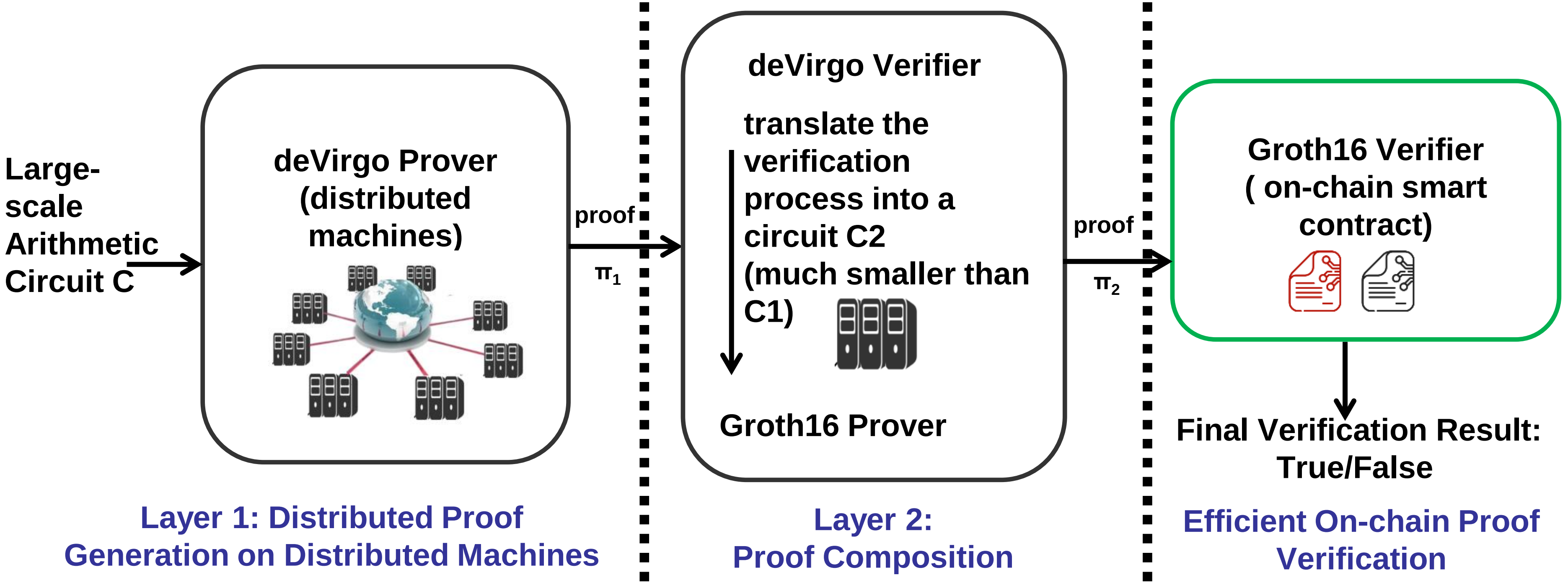
- deVirgo: a distributed version of Virgo (IEEE S&P 2020)
  - Exploits “data parallelism”
  - Optimal parallelization ---- 100x speedup with 128 machines
  - Practical communication ---- less than 20% of proving time
- Reducing proof size by recursion
  - run deVirgo verifier in Groth16
- Batching

deVirgo: **fast** proof generation, **relatively big** proof

Groth16: **slower** proof generation, **constant** proof & verification.

Constant size proofs & verification with only a slight increase in prover time

Approach: deVirgo & 2-layer Proof Composition



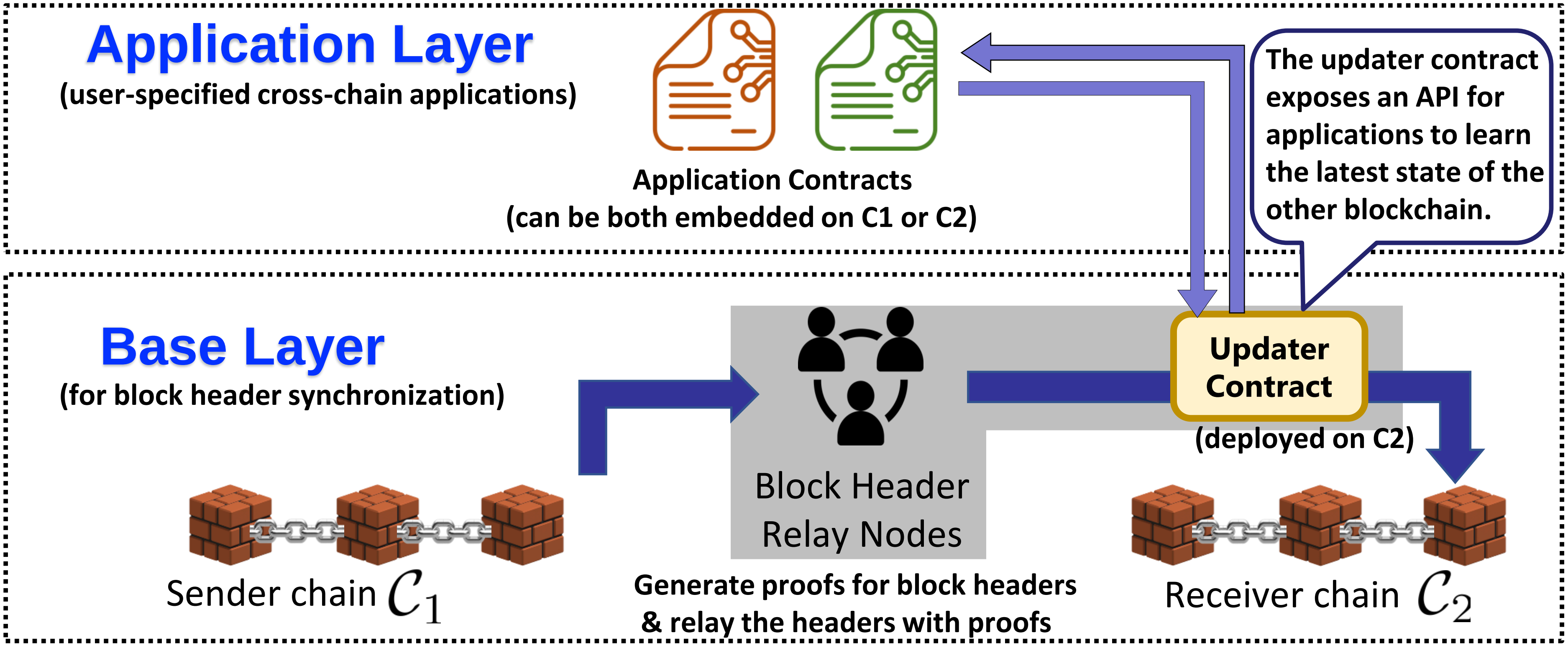
Performance of zkBridge proofs

	Proof Gen. Time (seconds)			Proof Gen. Comm. (GB)		Proof Size (Bytes)		On-chain Ver. Cost (gas)	
# of sigs	deVirgo	RV	total	total	per-machine	w/o RV	w/ RV	w/o RV	w/ RV
8	12.52	4.90	17.42	7.34	0.92	1946476	131	78M	221K
32	12.80	5.41	18.21	32.24	1.01	1952492	131	78M	221K
128	13.28	5.49	18.77	131.89	1.03	1958508	131	79M	221K

Table 2: Evaluation results. RV is the shorthand for recursive verification.

More results in paper: https://zkbridge.org.

Extensibility of zkBridge



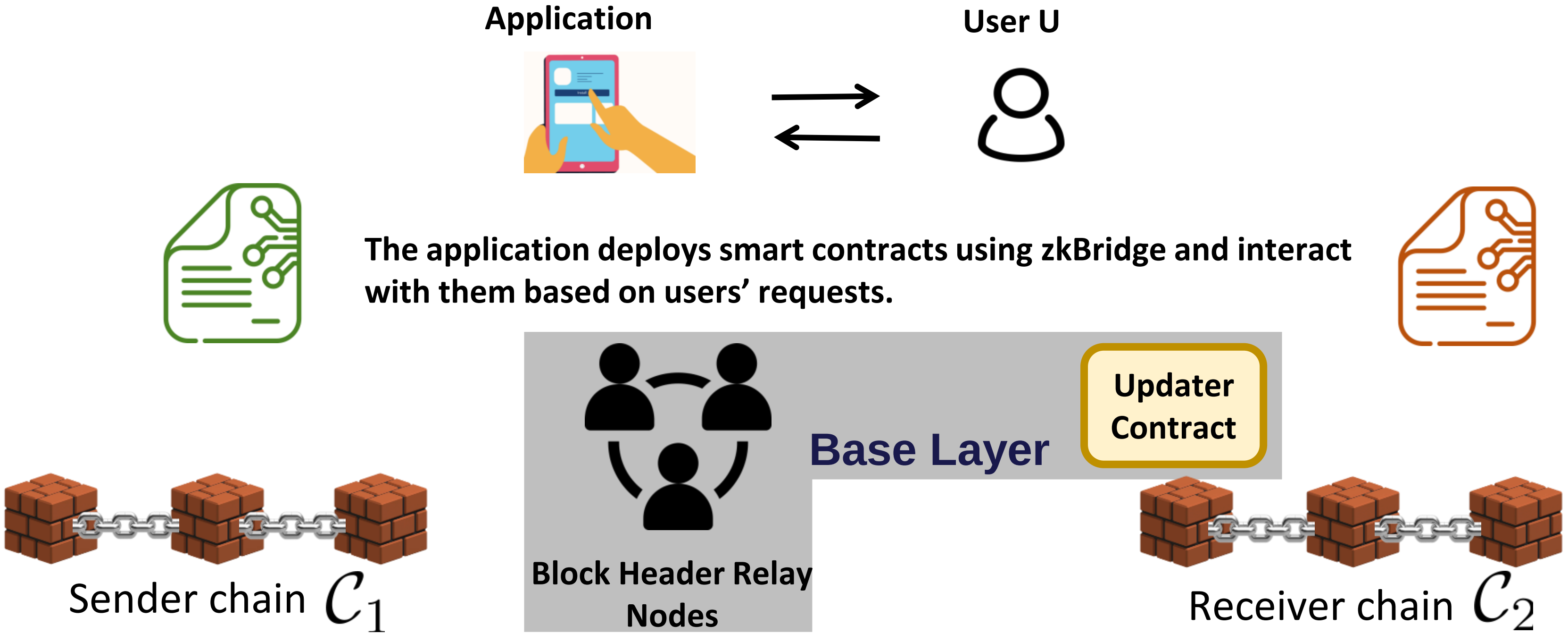
Extensibility & Applications

zkBridge has great extensibility

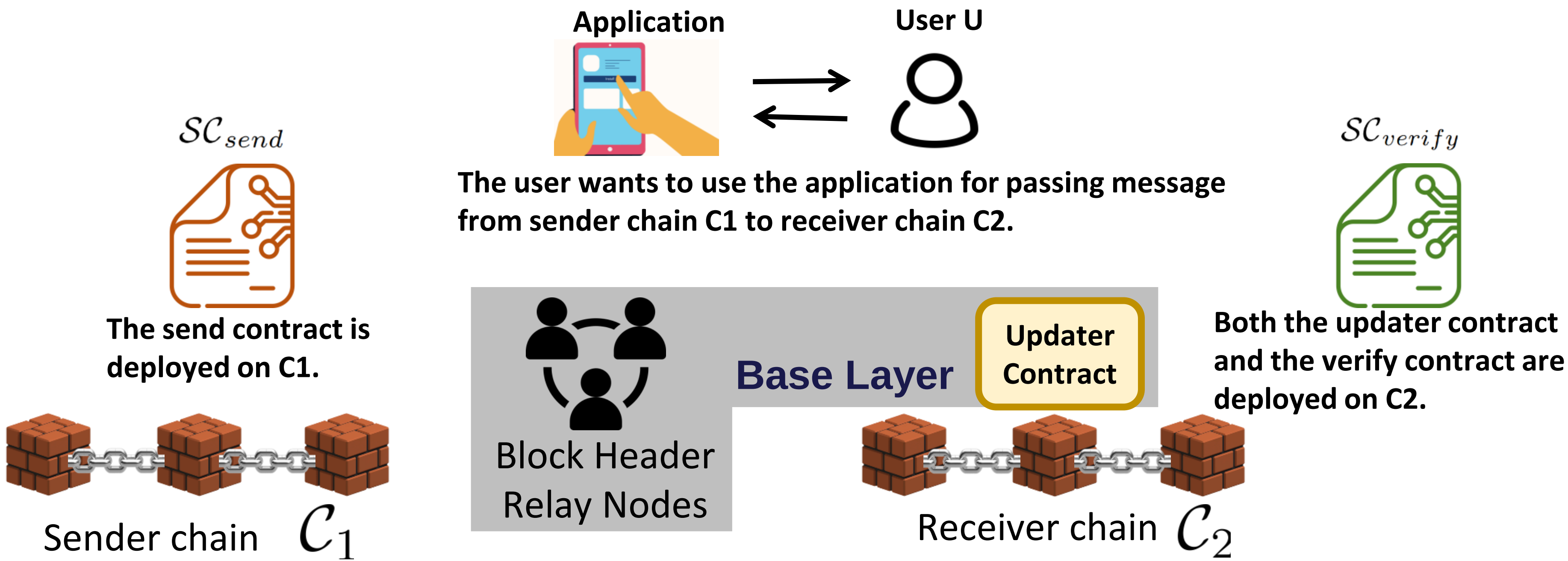
Developers can build application contracts to achieve more advanced functionalities such as:

1. Message Passing
2. Cross-chain Assets Transfer/Swap
3. cross-chain NFT Interoperations

Application Layer Components

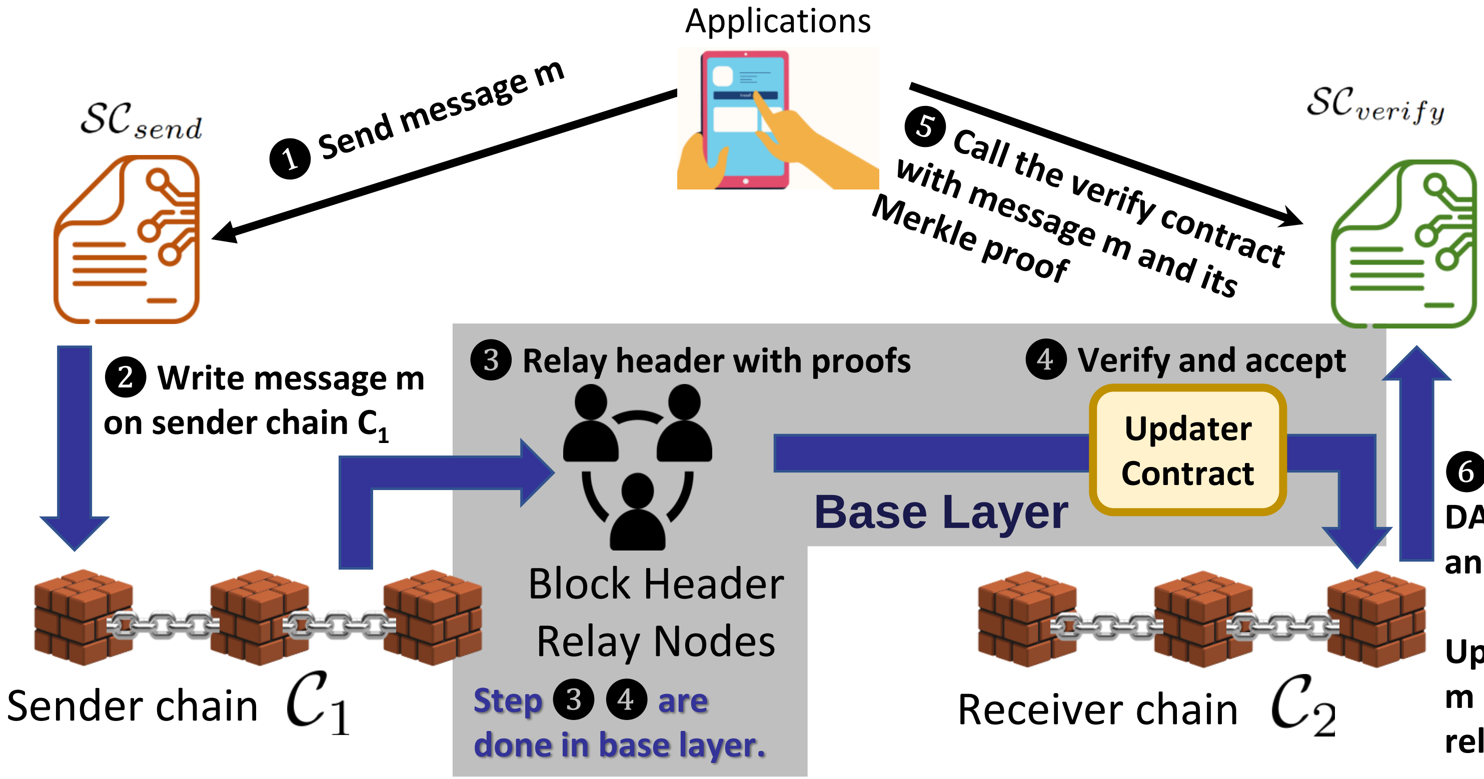


Application Layer Use Case 1: Message Passing



SC_{send} and SC_{verify} provides a message-passing service that other applications can use

Application Layer Use Case 1: Message Passing



Verification details:

❶ Identify the block containing m and verify the block is on the main chain of DAG;

❷ Verify submitted Merkle proof of message m in the corresponding block.

❻ Retrieve block header DAG from updater contract and verify message m.

Upon verification, message m is made available to relevant applications on C_2.

Defense-in-Depth

- Base layer of zkBridge presents a unified interface for syncing block header from another chain
- Improving security with defense-in-depth
  - Combining multiple implementations: proof-diversity, n-version programming, combining with other approaches such as optimistic solutions
  - Design different policies for combining different implementations
    - E.g., Hashi (https://github.com/gnosis/hashi): an EVM Header Oracle Aggregator

zkBridge: trustless bridge made practical

- Minimized trust
- Efficient on-chain verification
- Efficient proof generation
- Permissionless & decentralized by design
- Extensible and universal
- To learn more: **https://zkbridge.org, https://rdi.berkeley.edu/research**

Berkeley Center for Responsible, Decentralized Intelligence

- Tiancheng Xie, Jiaheng Zhang, Zerui Cheng, Fan Zhang, Yupeng Zhang, Yongzheng Jia, Dan Boneh, Dawn Song, “zkBridge: trustless bridge made practical”, ACM CCS 2022

zkBridge Technology Enables Other Capabilities

- State proof
  - A cryptographic proof of state changes that occur in a given set of blocks (e.g., Algorand State Proof)
- zk-based light client verification
  - Support efficient light client verification, including mobile use case (e.g., Celo Plumo)
- zkBridge can be extended to privacy chains with privacy protection

zkBridge Track in ZKP/Web3 Hackathon