Writing Smart Contracts
06 Smart Signatures

Peter H. Gruber

Supported by the Algorand Foundation

Smart signatures

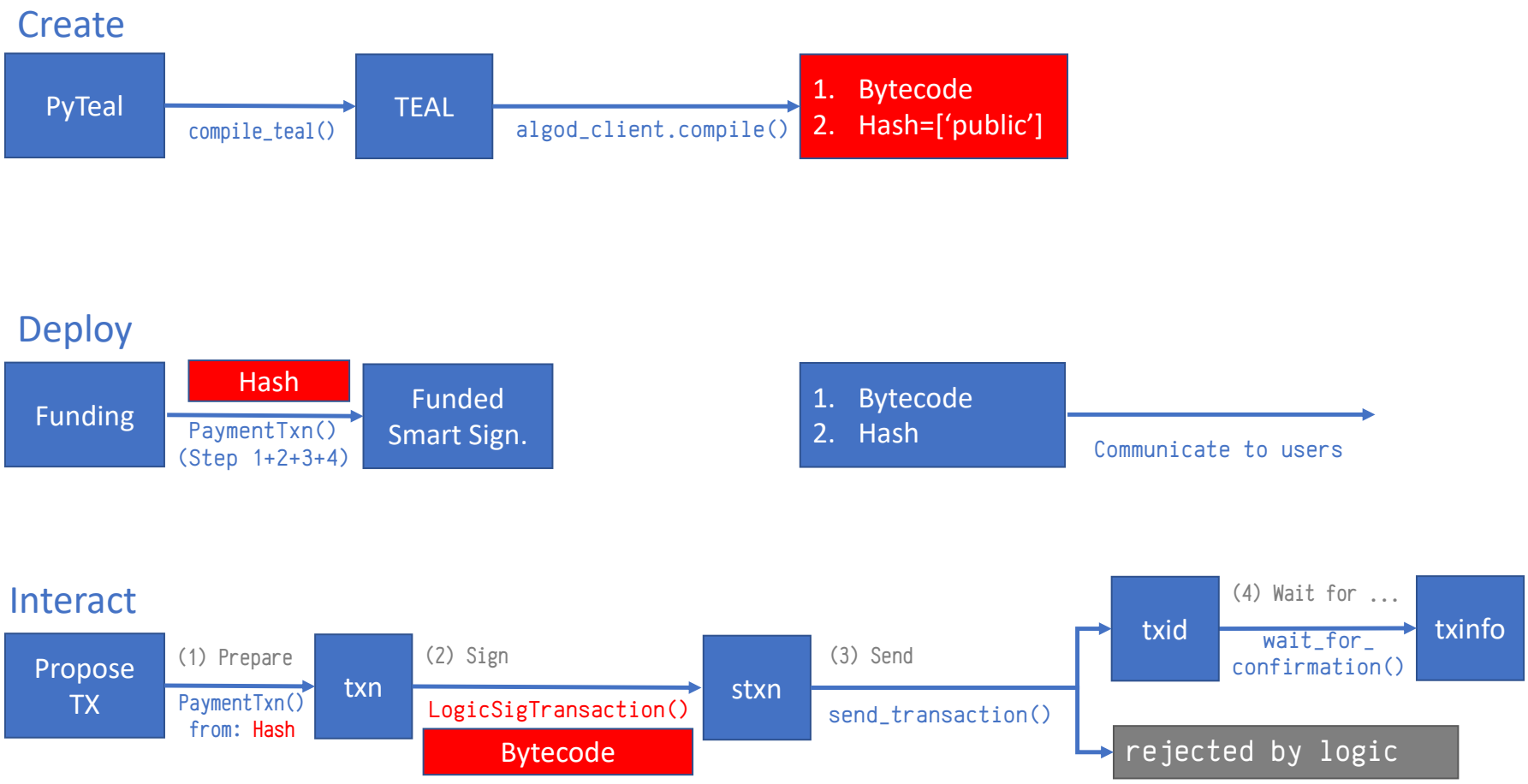
- A "yes" or "no" smart contract
  - Function that returns `True` or `False`
- CAccept or reject a **proposed transaction**

Arguments

- **Possible:** any property of the proposed TX
  - Receiver
  - Amount
  - First/last round
  - TX type (Payment or AssetTransfer)
  - Asset index
  - TX note, TX arguments
- **Not possible:** on-chain information
  - Asset holdings of smart signature or receiver
  - Transaction history

`https://pyteal.readthedocs.io/en/stable/accessing_transaction_field.html`

Life of a smart signature



Deployment of a smart signature

Step 1: Funding

- Public address = hash calculated from program code
- Transfer ALGOs to public address
  - Anyone can fund a smart signature
  - Multiple funding operations possible

Step 2: Communication

- Communicate to possible users of the smart signature:
  - Hash = public address
  - Result = program code
- Provide template (web) for acceptable transactions

Smart signature attacks

- Transaction fee attack
  - Propose transaction with excessive fee
  - Make smart sig loose money
- Rekey attack
  - Rekey smart sig to other address
  - Steal all money from smart sig
- Closeout attack
  - Add closeout to valid payment transaction
  - Steal all money from smart sig
- Cross-asset attack
  - Propose attack for a different ASA as intended
  - Obtain ASA much cheaper than intended
- Transaction group attack