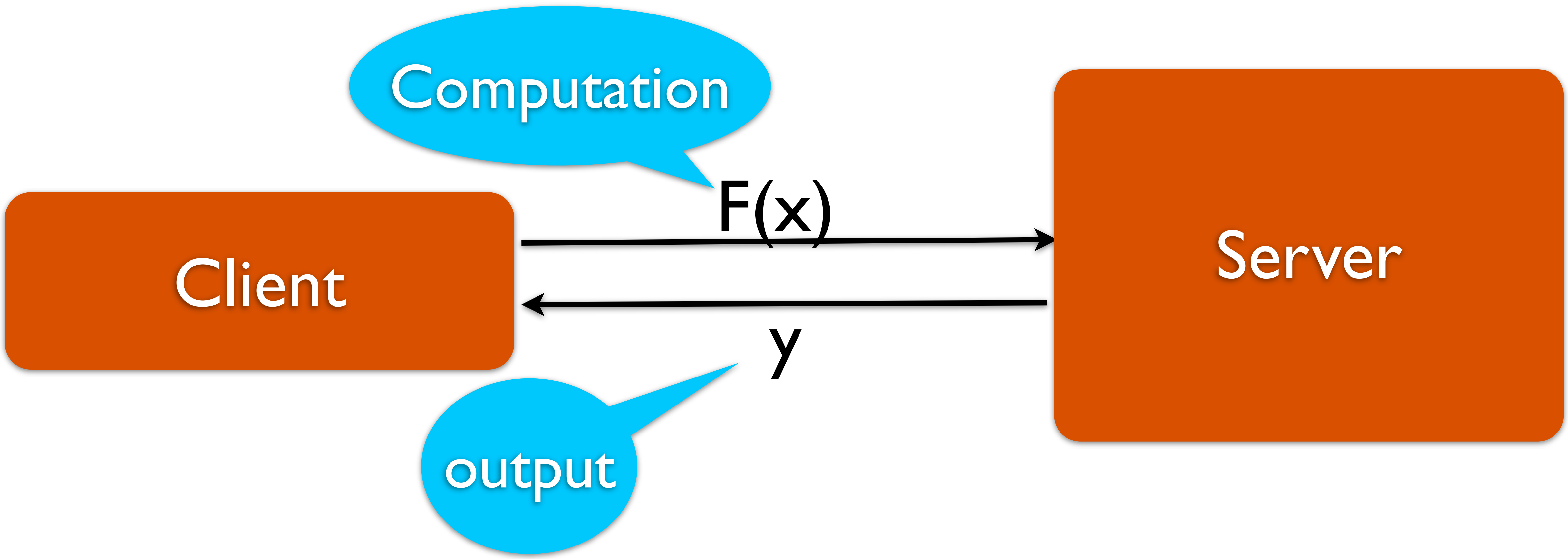
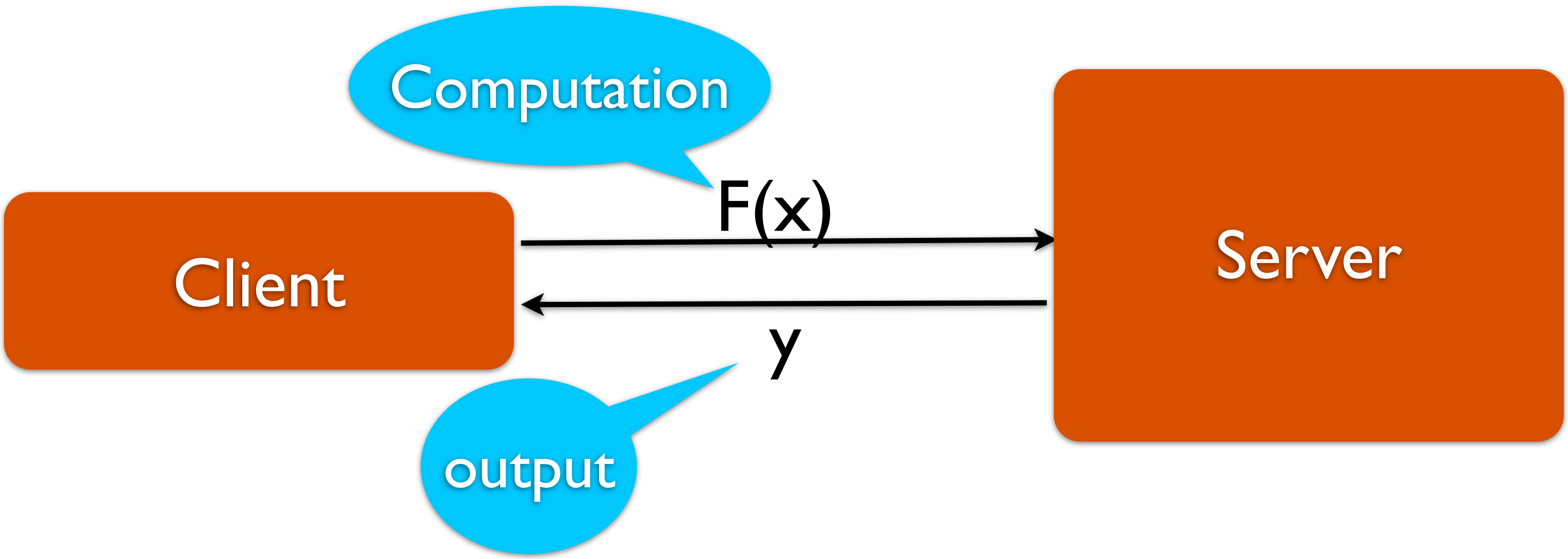
Verifying remote computations using PCPs

Srinath Setty, Andrew Blumberg, and Michael Walfish
UT Austin

Can we build this?

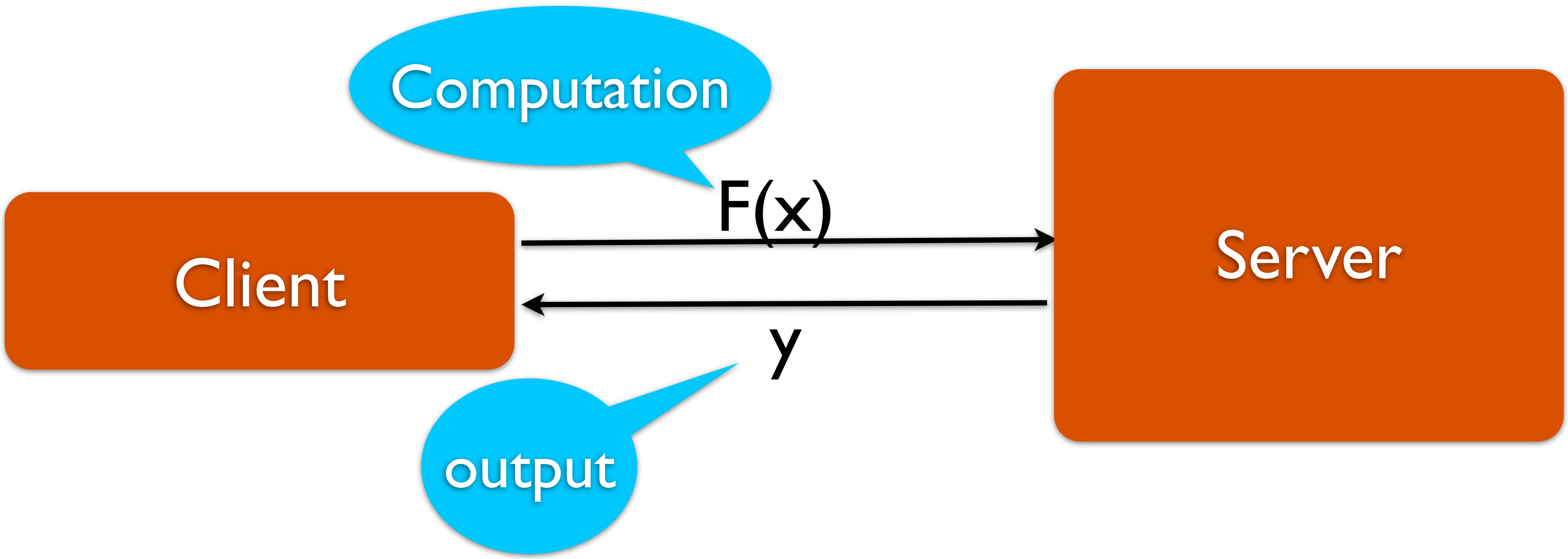


Can we build this?



- Check if y equals F(x) without re-executing

Can we build this?



- Check if y equals F(x) without re-executing
- Unconditional: no assumptions

Why should we build this?

- Offloading computations to the cloud
- Outsourcing computations to volunteer machines (Enigma@home, Einstein@home, ...)

How can we solve this problem in principle?

- Probabilistically checkable proofs (PCPs) and argument systems [Arora et al. JACM, 1998]

How can we solve this problem in principle?

- Probabilistically checkable proofs (PCPs) and argument systems [Arora et al. JACM, 1998]

- PCP theorem: server proves that y = F(x) and client validates without re-executing

We have a conflict

- PCPs are mind-blowing

We have a conflict

- PCPs are mind-blowing
- But the costs are also mind-blowing

We have a conflict

- PCPs are mind-blowing
- But the costs are also mind-blowing
  - For polynomial evaluation (700 variables), the server takes 10^5 years!

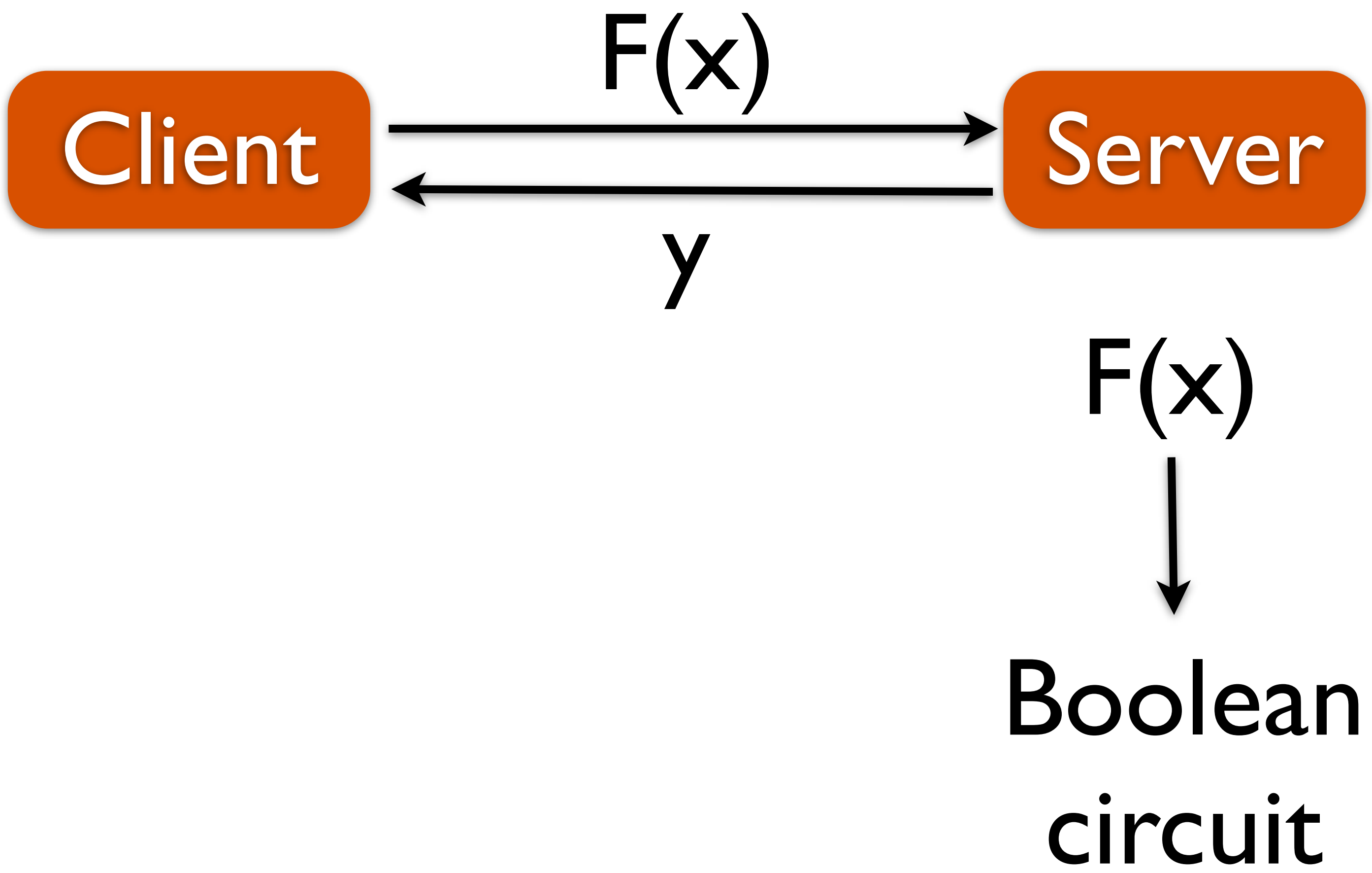
We have a conflict

- PCPs are mind-blowing
- But the costs are also mind-blowing
  - For polynomial evaluation (700 variables), the server takes 10^5 years!
- Our research program: try to make PCPs practical

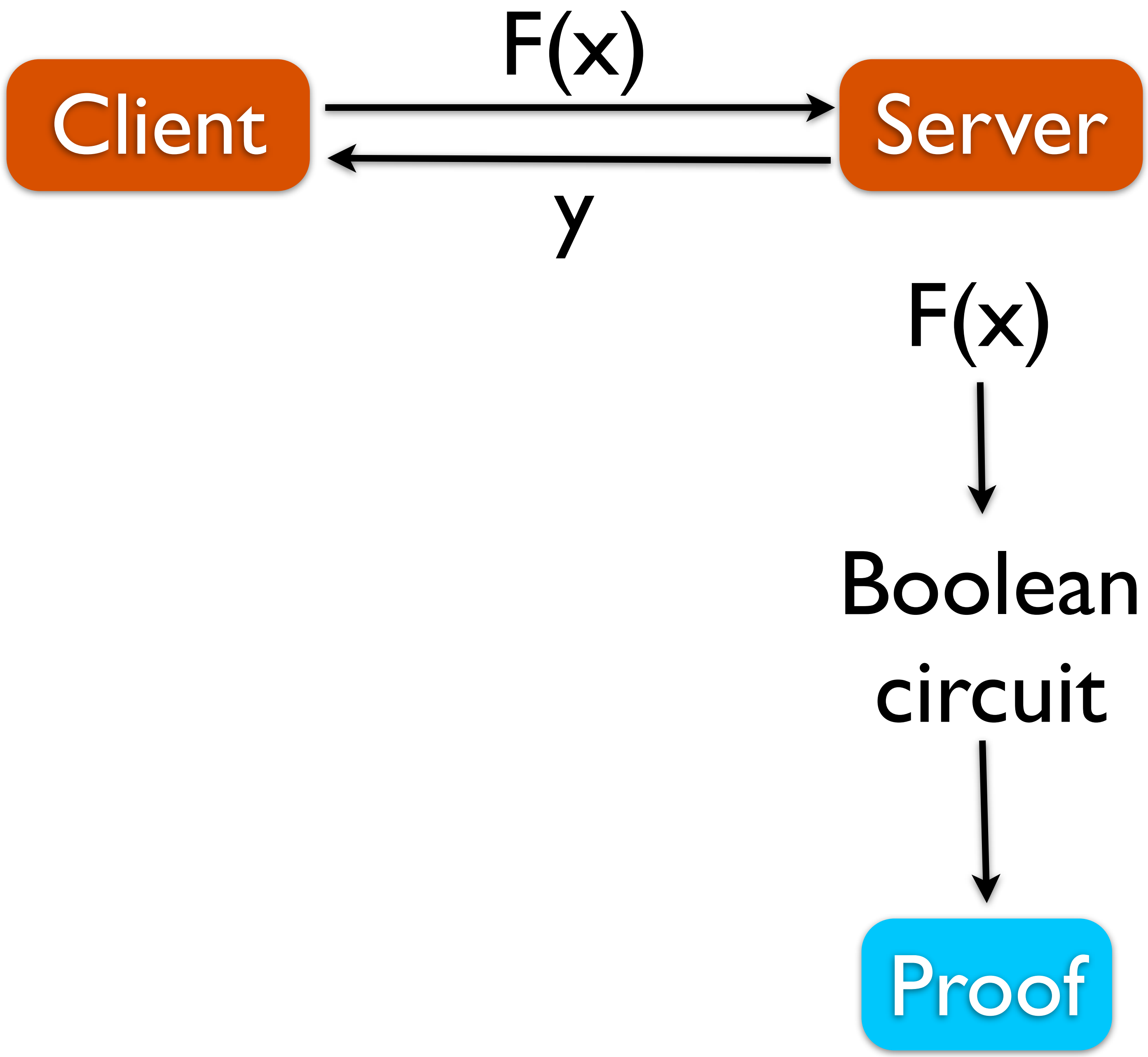
Rest of this talk:

- Overview of PCPs
- Our refinements

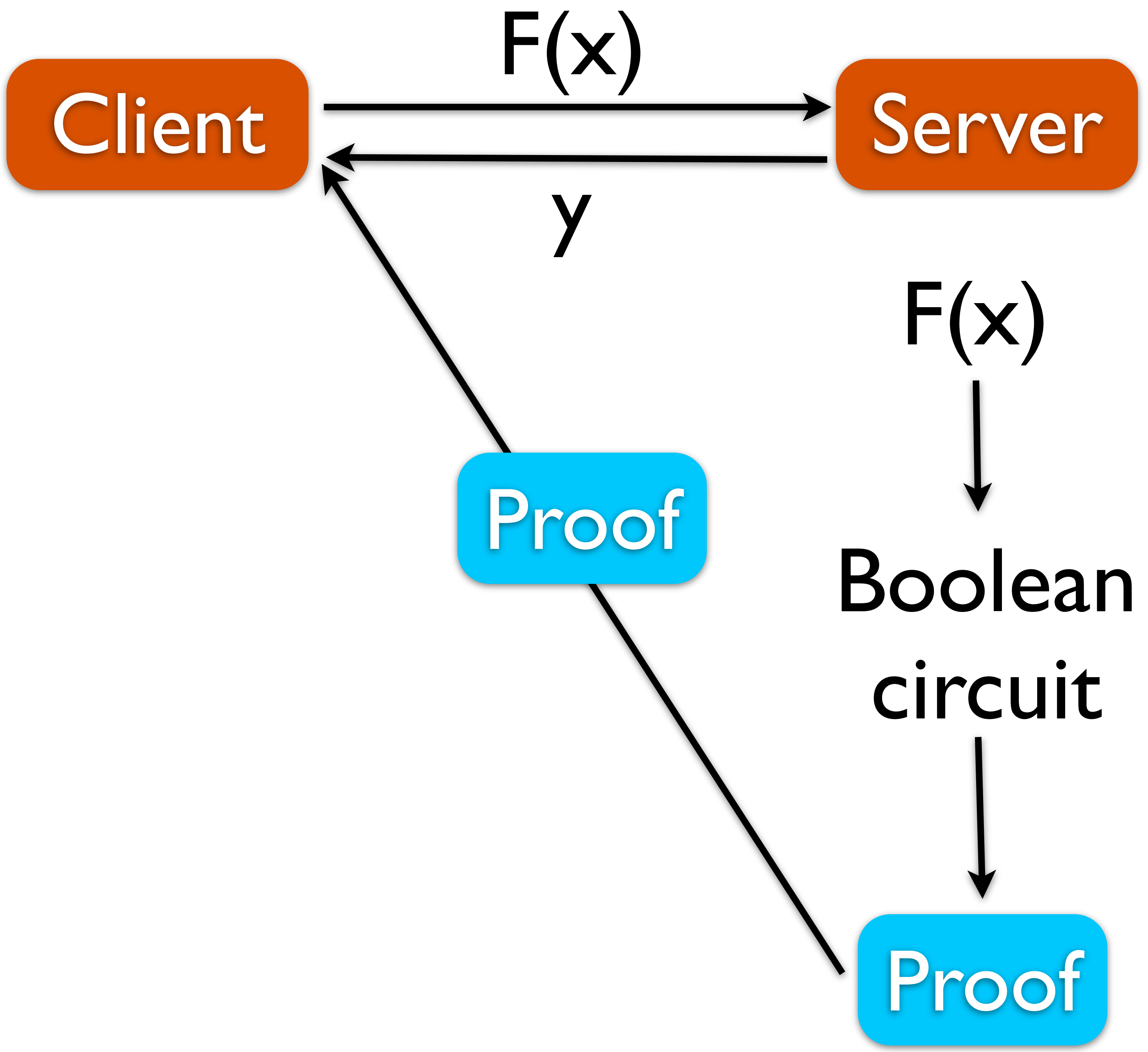
PCPs from 200,000 feet



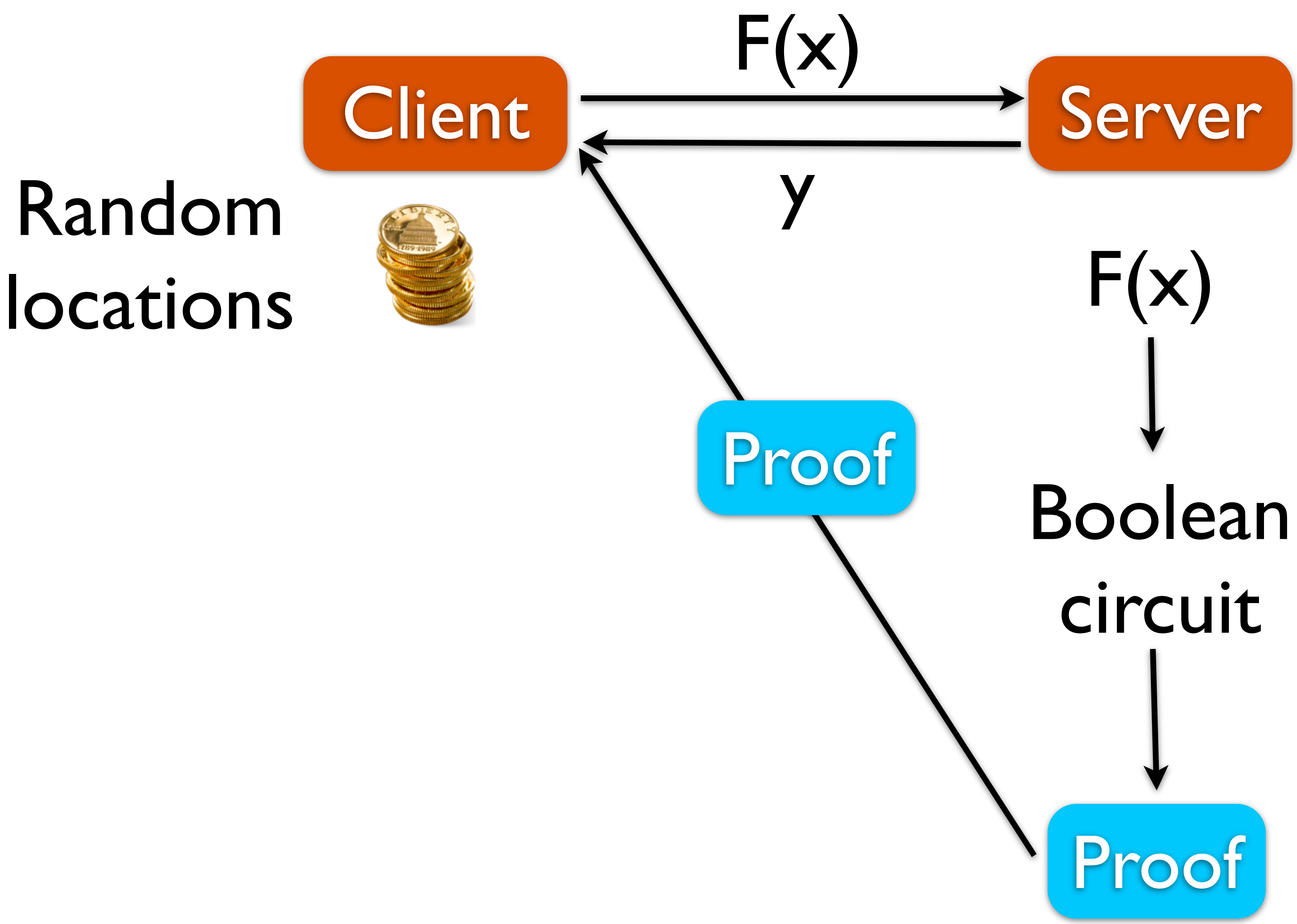
PCPs from 200,000 feet



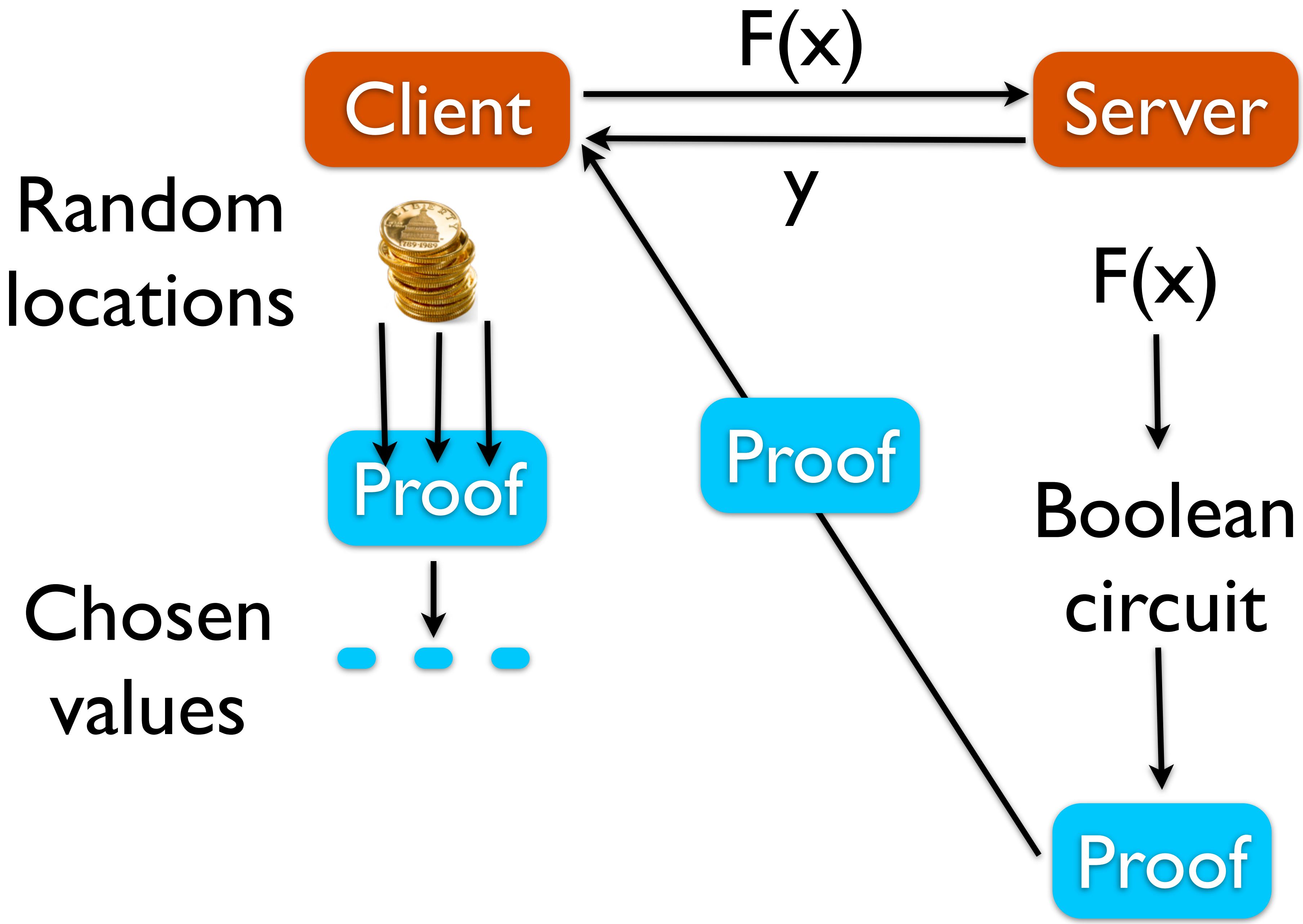
PCPs from 200,000 feet



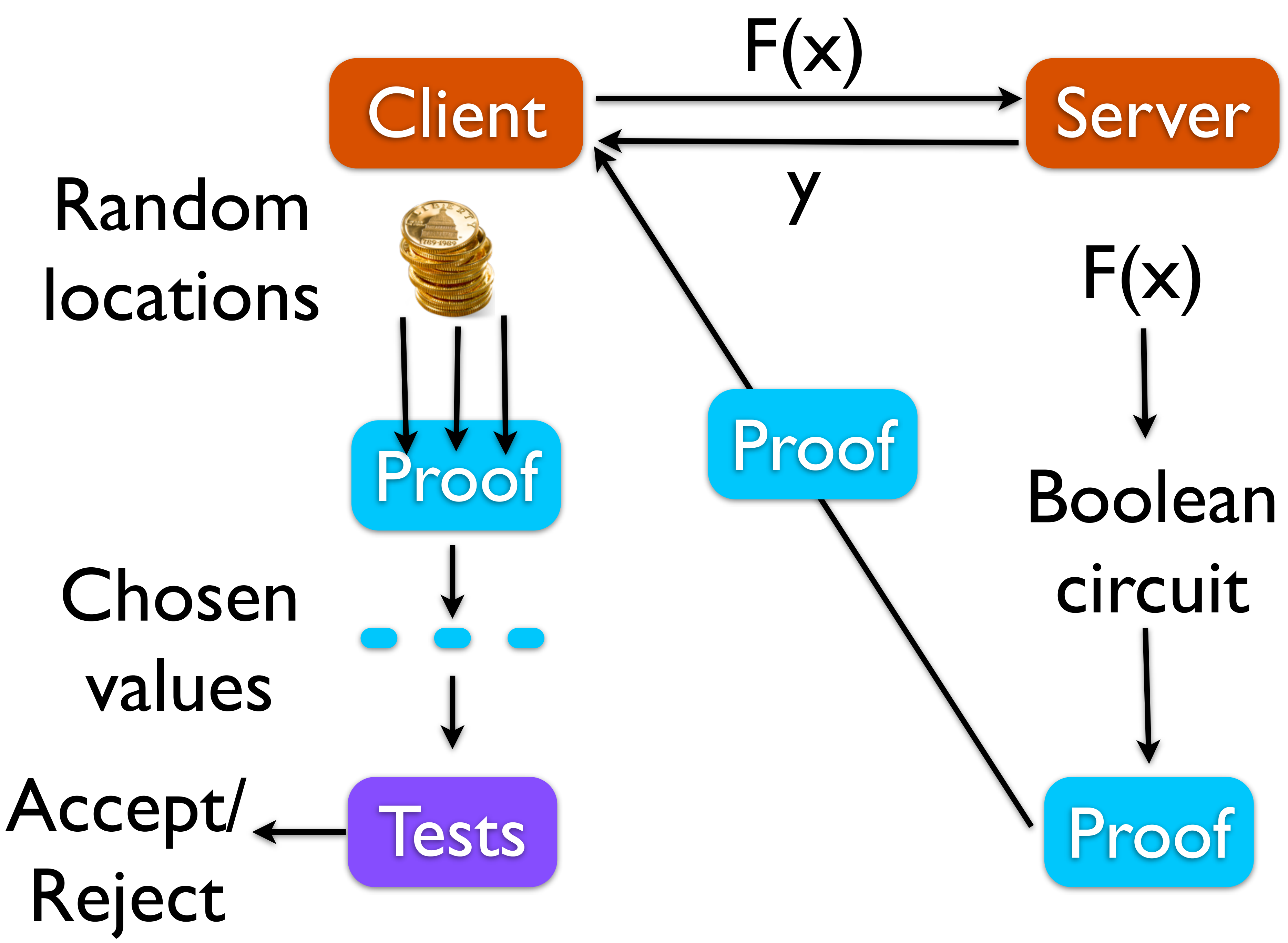
PCPs from 200,000 feet



PCPs from 200,000 feet



PCPs from 200,000 feet



Our attempt to make PCPs practical

- Build on the work that introduces interaction [Kilian CRYPTO'95, Ishai et al. CC'07]
- Use a higher-level abstraction to represent computations
  - Reduces cost by 8 orders of magnitude
- Apply a divide-and-conquer technique
  - Reduces cost by 2 orders of magnitude

We build on an interactive variant of PCPs

[Ishai et al. CC'07]

- The server proof is a generating function
- The server responds to queries by evaluating the function
- The client binds the server to its function using cryptographic commitment

Can we use a higher-level abstraction?

- Use arithmetic circuits instead of Boolean circuits
- Savings:
  - 8 orders of magnitude at the server
  - 4 orders of magnitude at the client

Can we apply a divide-and-conquer strategy?

- Decompose the computation into parallel pieces
- The client batch-verifies the computation
- Saves two orders of magnitude in costs

Examples that we implemented

- Polynomial evaluation
- Matrix multiplication
- Fast Fourier Transform (FFT)
- Image filtering with convolution matrices

Example savings

For polynomial evaluation with 700 variables

	interactive baseline	post-refinements
Server's work	130,000 years	11.5 hours
Client's work	940 sec	94 msec

(Local execution time: 164 msec)

→ The scheme is near-practical

Summary

- Our refinements reduce costs by over 10 orders of magnitude
- More refinements are required to make the scheme fully practical
- Upshot: PCP-based verified computation can be a systems problem