Multiparty System Governance

and the Shared Signals Use-Case

Thomas J. Smedinghoff
Locke Lord LLP

Open Identity Exchange Blockchain, Identity, Trust, and Governance (BITGov) Workshop
Stanford, May 9, 2018

Focus is on Multi-Party Systems

Examples of Multi-Party Systems

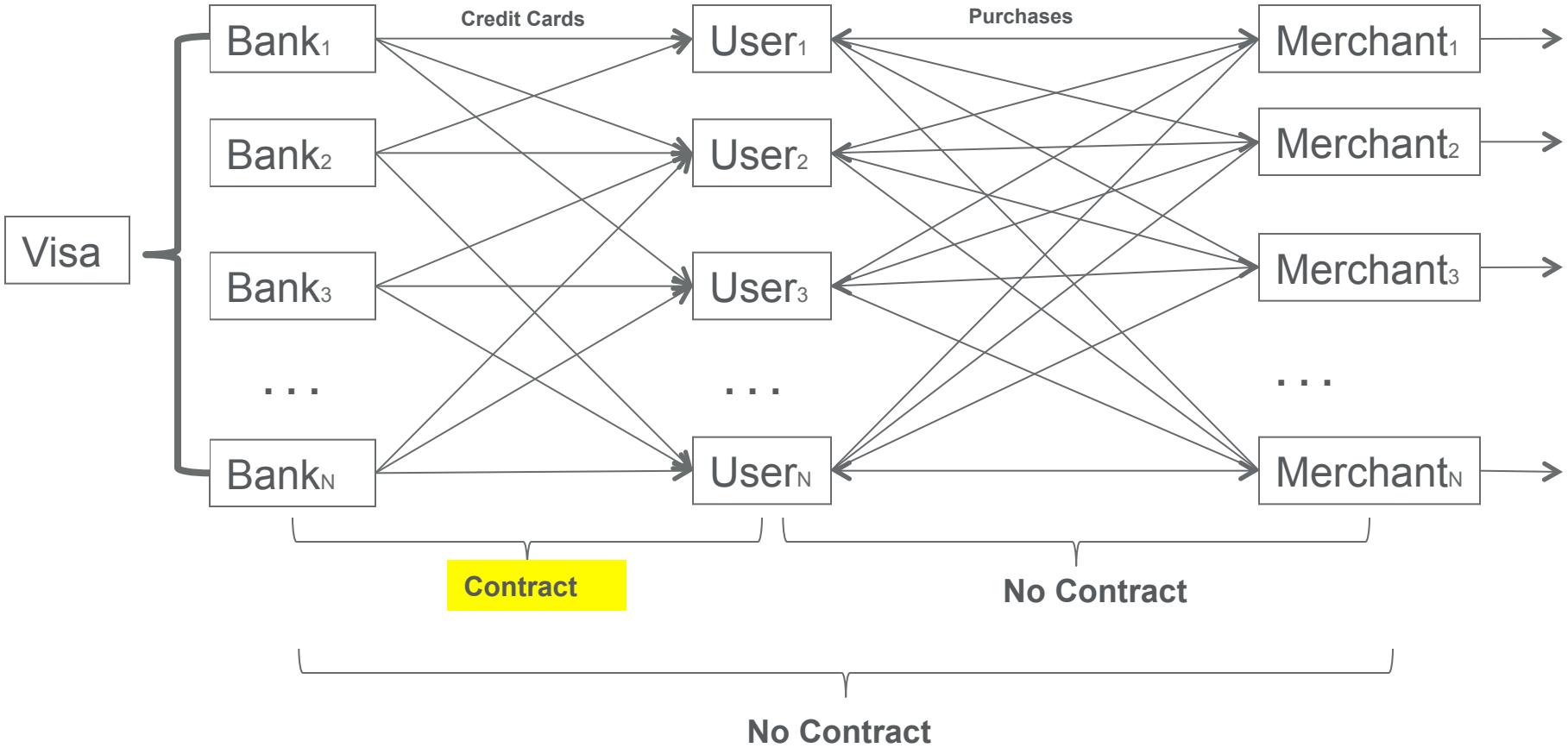
- **Credit card systems**
  - American Express
  - Discover
  - MasterCard
  - Visa
- **Payment systems**
  - ACH
  - SWIFT
  - ATM systems
- **Identity systems**
  - SAFE-BioPharma (pharma sector)
  - IdenTrust (financial sector)
  - InCommon (education sector)
  - Gov.UK Verify (UK residents)
  - eIDAS (EU residents)

Characteristics of a Multi-Party System

- **Numerous participants**
  - Potentially thousands or millions
- **Common transaction type** or transaction set
  - Credit card transactions; payment transactions; identity transactions; stock transfer transactions;
- ** **Random interaction among participants** **
  - Any participant can transact with any other participant
- **Distributed processing**
- **Cross jurisdictional reach**
- **Self-regulated** (ideally)

Interactions of a Multi-Party Credit Card System

Data Flow: Multiple Banks issue credit cards to multiple cardholders (Users) who use those cards to make purchases from multiple Merchants



Multi-Party System Governance

All Multi-Party Systems Need . . .

- Rules
- A mechanism to make rules enforceable on the participants
- A governance mechanism
  - To make the rules
  - To amend / maintain the rules

All Multi-Party Systems Need Following Types of Rules

- Operational rules
  - How is the system supposed to work?
  - What are the processes used?
- Technical rules
  - How is the data structured, formatted, communicated, secured, verified, etc.?
- Business rules
  - Who can (or is supposed to) do what?
  - What roles exist and what are their duties and responsibilities?
- Legal rules
  - Compliance requirements
  - Risk and loss allocations
  - Warranties and liability

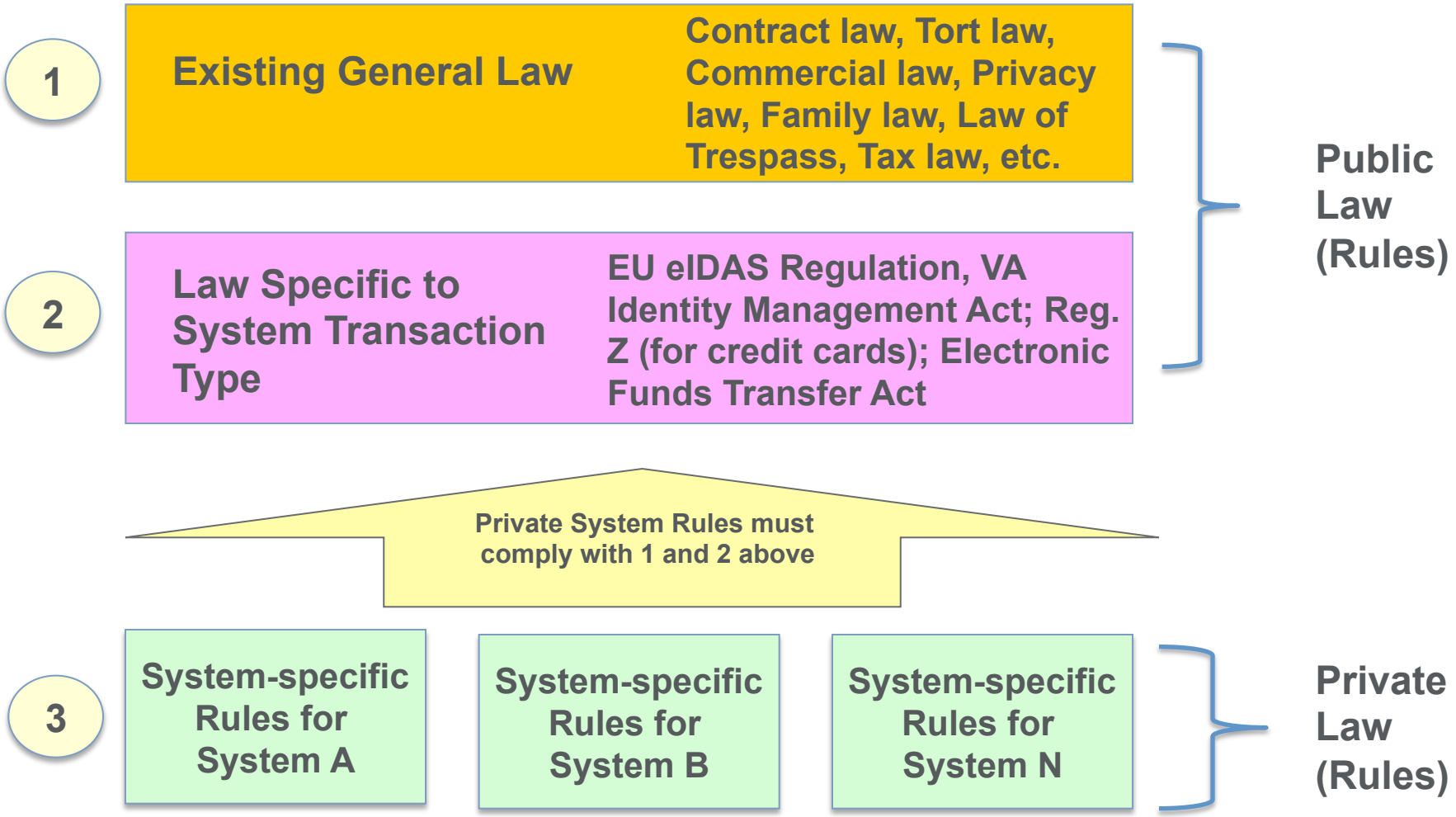
Purpose of Rules for Multi-Party Systems --

- Make the system "operationally functional"
  - So that it "works"
  - So that everyone knows what to do / how to design it, etc.
- Make the system "trustworthy"
  - Goes beyond merely functional
  - Address and minimize the risks
  - So that participants have confidence in the results and are willing to rely on them
- Address the legal issues
  - Define participant legal rights, duties, and obligations
  - Clearly define and fairly allocate liability risks
  - Fill gaps not covered by law
  - Resolve ambiguous law (e.g., who is liable if "x" happens)
  - Alter inconsistent law (where allowed)

Where Do the Rules Come From? Three Basic Sources -

1. Public law – General law
   - Existing statutes, regulation, and case law
   - Not written to address specific system transaction issues
   - E.g., contract law, tort law, privacy/security law, commercial law, personal injury law, family law, tax law, competition law, etc.
2. Public law – Law specific to Transaction Types
   - Written to address the category of system transactions; apply to all systems of that type
   - E.g., EU eIDAS Regulation, Virginia Electronic Identity Management Act; Reg. Z (for credit cards); Electronic Funds Transfer Act, etc.
3. Private law – Contracts or ??
   - Often called **trust frameworks**, **scheme rules**, or **system rules**
   - Written to govern a specific identity system
   - E.g., Visa rules, ACH rules, SAFE-BioPharma rules, etc.

Multi-party Systems: Three Levels of Rules Can Govern



Mechanism to Make the Rules Enforceable

- Levels 1 and 2
  - By law
- Level 3
  - Binding contracts
  - Other possibilities for selected rules
    - Custom & usage (in certain situations)
    - Code (software, etc.)
    - Peer pressure

Possible Governance Mechanisms

- Independent entity formed for the purpose of governing the system
  - E.g., Visa, NACHA
- Controlling participating entity
  - E.g., UK Cabinet Office for UK.GOV Verify
- Consortium of participating entities
  - E.g., group of banks
- Informal committee
  - E.g., InCommon Steering Committee, policy committee, etc.
  - Perhaps elected or appointed by participants
- Vote of all participants
- Ad hoc – e.g., each participant handles independently

Shared Signals System Use Case

What Are Shared Signals?

- User account events of potential interest to other entities
  - E.g., To assist in detecting fraud
- Account events include –
  - Account Credential Change Required (e.g., new password)
  - Account Purged
  - Account Disabled
  - Account Enabled
  - Identifier Changed (e.g., name change)
  - Identifier Recycled (i.e., belongs to new user)
  - Recovery Activated
  - Recovery Information Changed
- Shared signals enables intelligence sharing between entities – i.e., sharing account event notifications

Technical Specifications Developed by OpenID Foundation

- OpenID Risk and Incident Sharing and Coordination (RISC) Working Group --
  - Developing shared signals specifications
  - Initial Focus -- Internet accounts that use email addresses or phone numbers as the primary identifier for the account
- Purpose
  - Share information about important security events in order to thwart attackers from leveraging compromised accounts from one Service Provider to gain access to accounts on other Service Providers (mobile or web app developers and owners).
  - Enable users and providers to coordinate in order to securely restore accounts following a compromise.

Some Potential Issues for Shared Signals System Rules

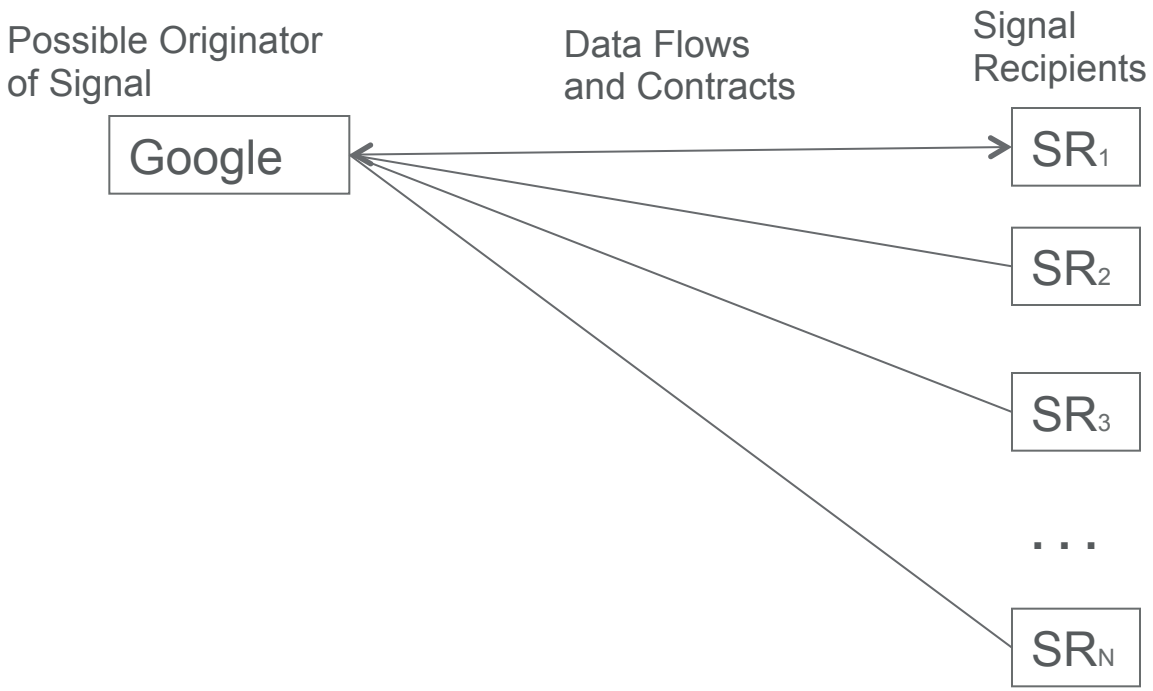
- Account event definitions – triggers, meaning, etc.
- Data formats, timing, etc.
- Privacy issues
- Security issues
- Responsibilities of issuing entity
  - E.g., Duty to send? Optional? Timing of sending
- Responsibilities of receiving entities
  - E.g., Right to rely? Duty to act?
- Confidentiality of information
- Right to share data with others
- Liability for bad data or unreasonable reliance
- Revocation of account event signal
- IP rights

Approaches to Shared Signals System Governance

- Bi-lateral Contracts Model
  - Every participant enters in to a contract with every other participant with whom they will transact
  - Rules are embodied in each contract

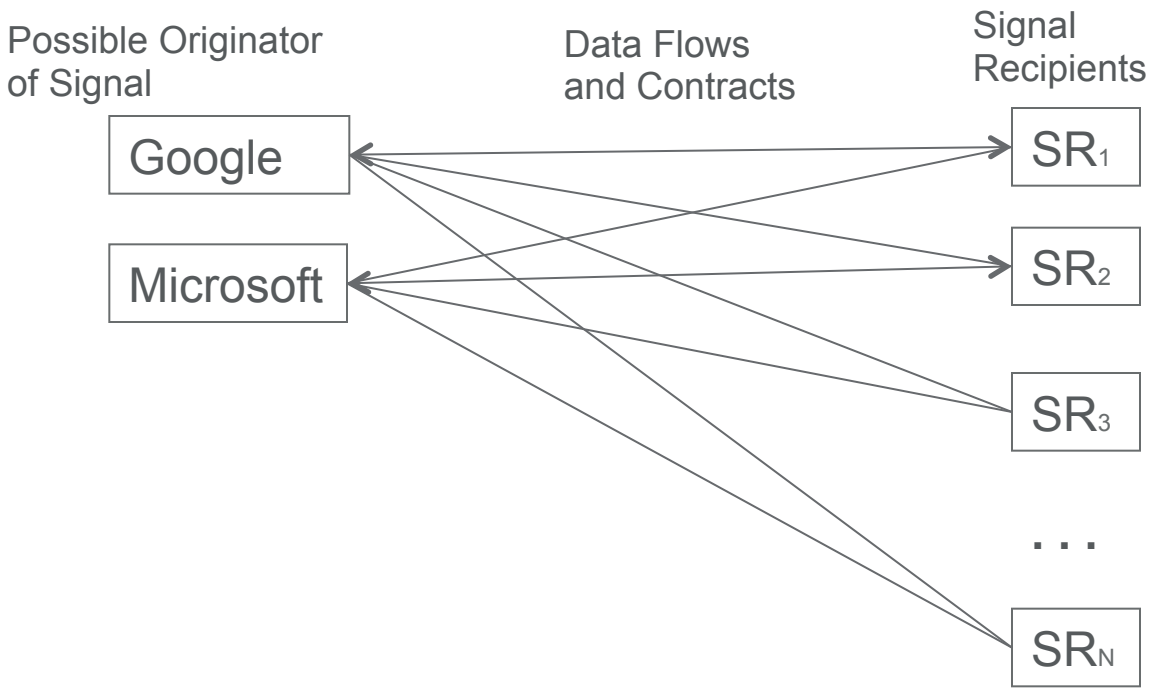
- Trust Framework Model
  - Every participant agrees once to a common set of rules (i.e., a trust framework) that is binding on all participants
  - Rules are embodied in a single Trust Framework

Shared Signals System
Bi-lateral Contract Governance



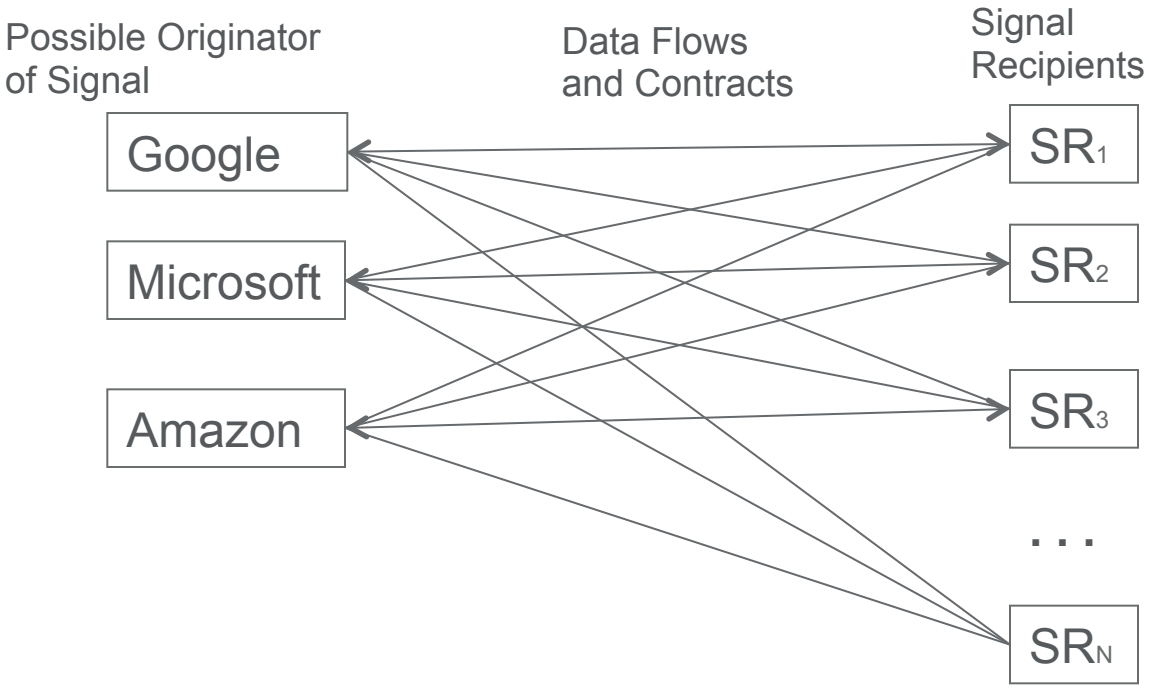
Data flows can go both ways

Shared Signals System
Bi-lateral Contract Governance



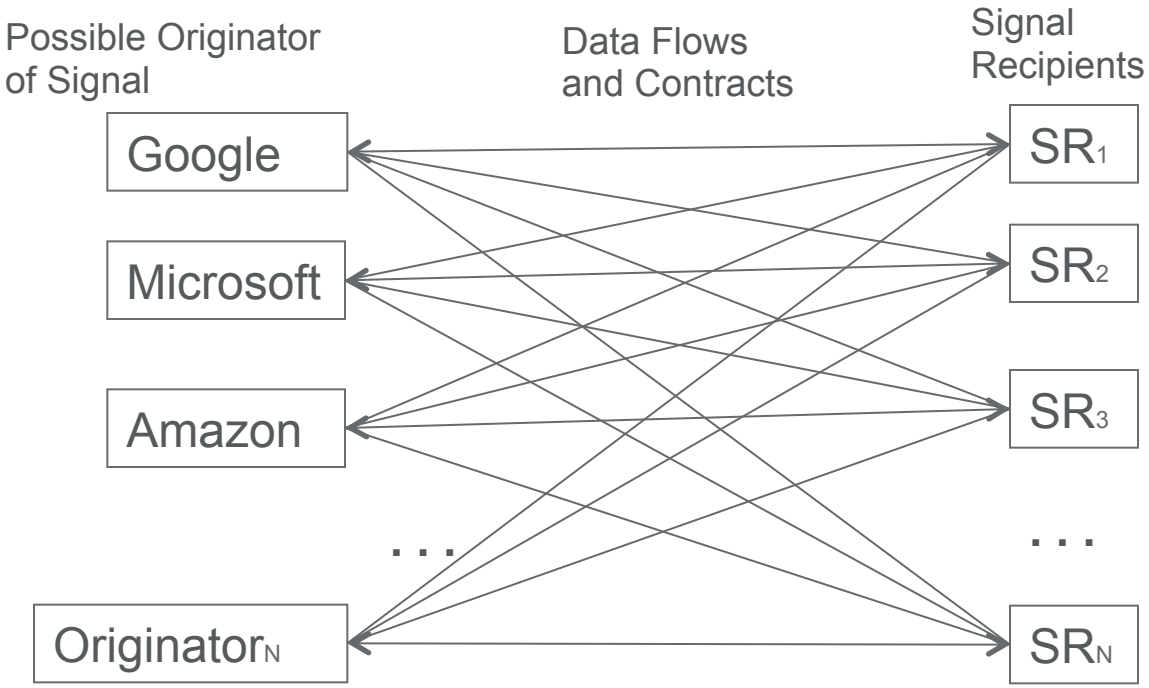
Data flows can go both ways

Shared Signals System
Bi-lateral Contract Governance



Data flows can go both ways

Shared Signals System
Bi-lateral Contract Governance



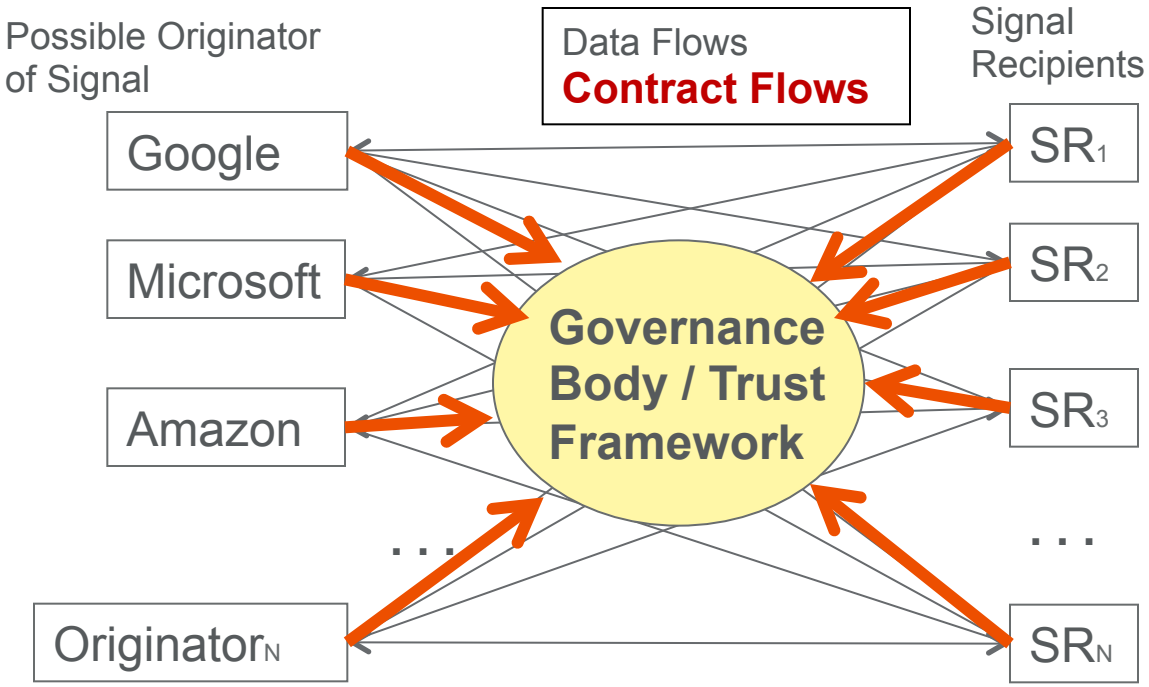
Data flows can go both ways

Bi-lateral Contract Governance

- Every participant enters in to a contract with every other participant with whom they will transact
- Easiest, quickest, & most economical to implement system
- Works relatively well in small multi-party systems

- But . . .
  - Requires contract between each transaction pair
  - Number of contracts grows exponentially as system scales
  - Each participant needs lots of contracts
  - Contracts may vary, so participants can't always assume each transaction governed by same rules
  - Rules changes may be impossible to implement universally
  - There is no trust in the overall system, only in individual contracts, which may vary

Shared Signals System Trust Framework Governance



Data Flows same as bi-lateral model; Contract flows simplified to one per participant

Trust Framework Governance

- Every participant agrees once to a common set of rules that is binding on all participants
- Same rules apply to everyone – can form basis of overall system trust
  - i.e., each participant knows how each other participant is obligated to act
- When necessary, rules can be changed for all participants uniformly

- But . . .
  - Much more work / greater cost to organize and set up