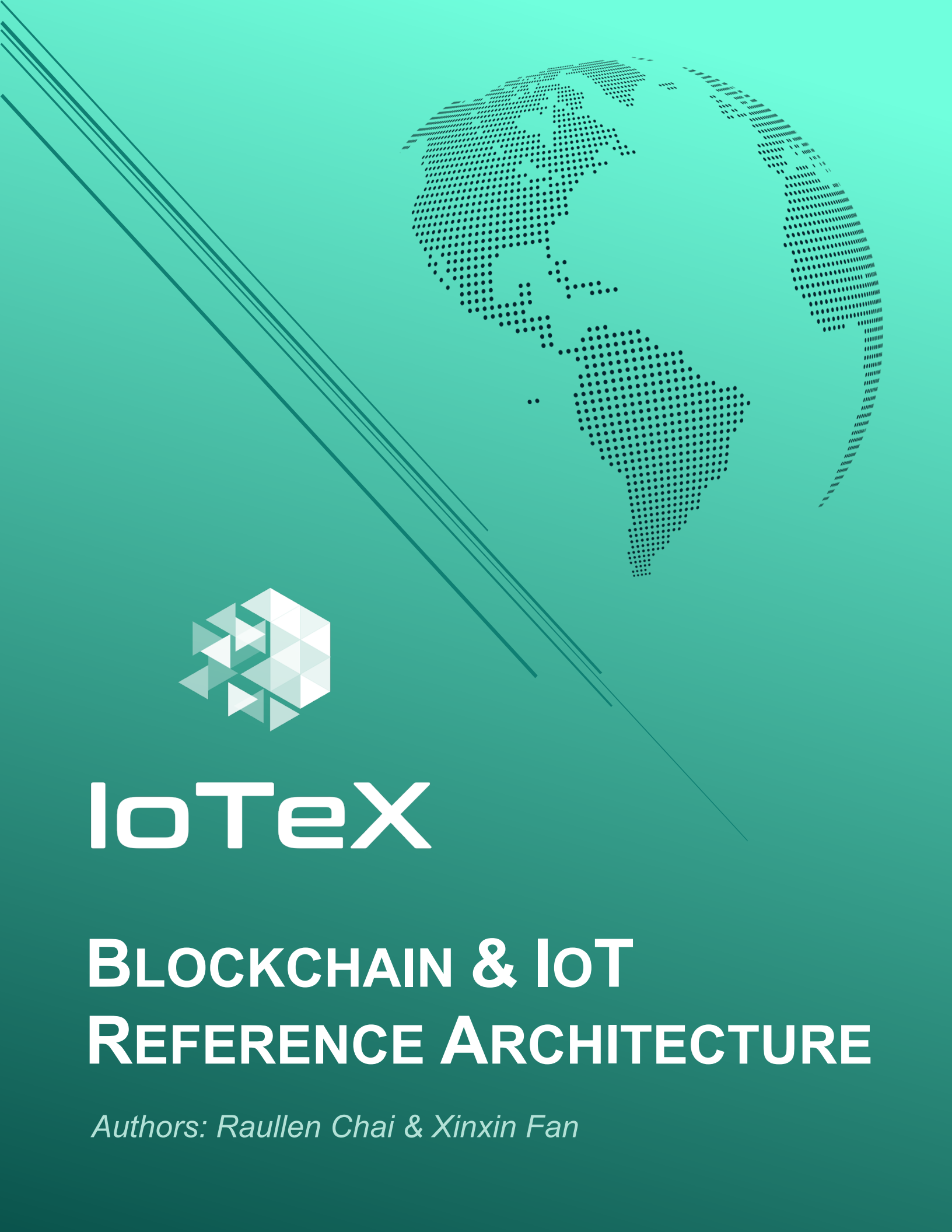


IoTeX

Blockchain & IoT Reference Architecture

Authors: Raullen Chai & Xinxin Fan

Reference Architecture Overview

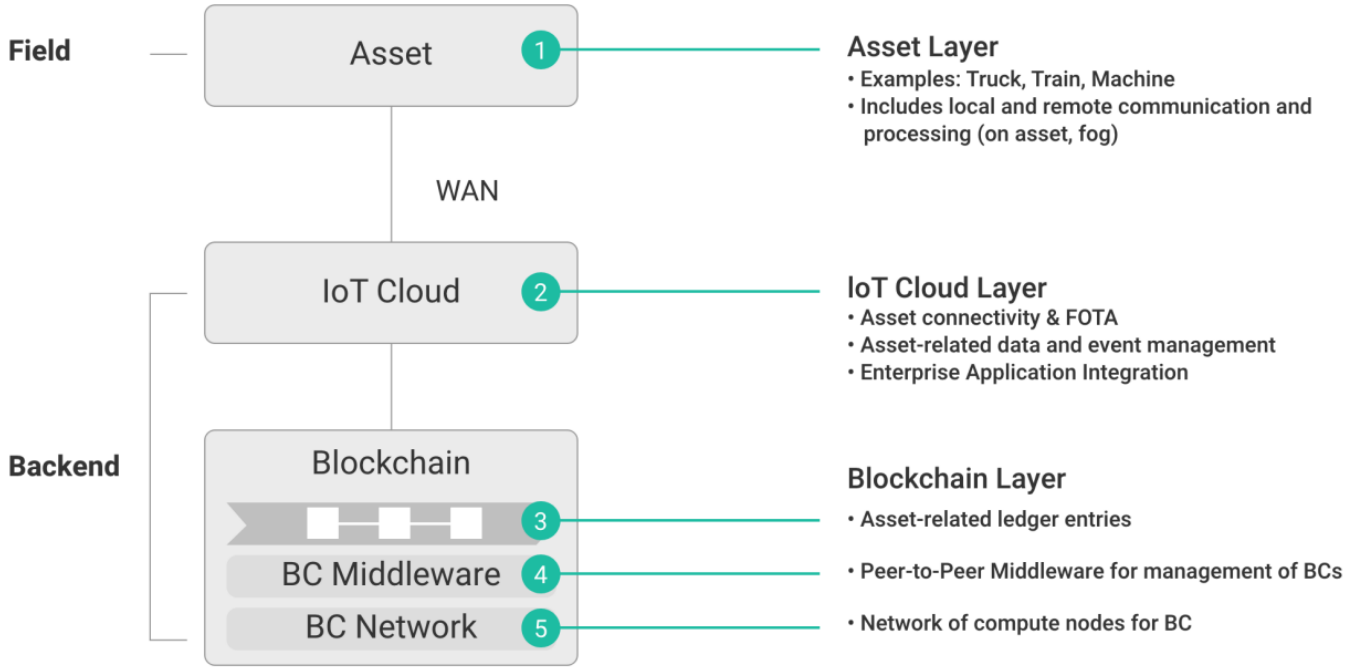


Figure 1: Reference Architecture Overview

Asset Layer

The asset layer represents a variety of physical assets in the field that are connected to the Internet via smart devices. These smart devices capture the physical properties of the associated assets and facilitate two-way communication and specific actions, as needed. The asset layer involves various IoT devices and gateways, as well as wireless communication protocols.

IoT Cloud Layer

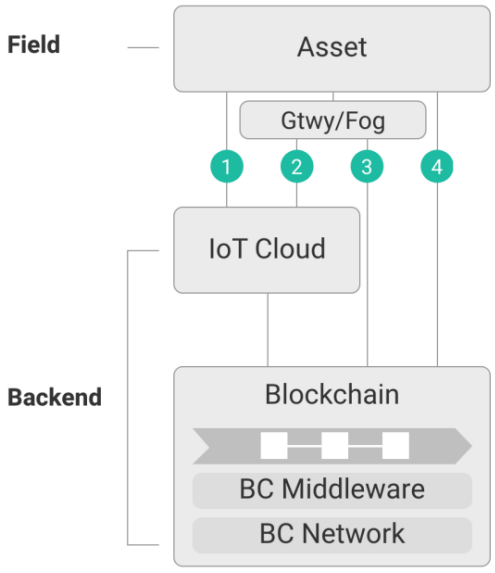
The IoT Cloud layer contains a complete set of components for connecting, processing, storing and analyzing asset-related data on the “edge” and in the Cloud. The core functionalities provided by this layer include device identity management, device connectivity management, device data storage, device data analytics, and device controls, as well as various automation and integration interfaces.

Blockchain Layer

The blockchain layer is the “root of trust” and is comprised of the following three sub-layers:

- **Applications:** maintains users’ accounts and records all transactions and blocks related to the physical assets
- **Middleware:** IoT-oriented software services to facilitate the creation, storage, and utilization of data from smart devices
- **Network:** comprises all the nodes that provide compute, storage, and network to support various blockchain operations, including consensus and P2P communications

Blockchain & IoT Integration Patterns



	a.) W-Only	b.) R-Only	c.) RW
1: Asset-to-Cloud-to-BC			
2: Gtwy-to-Cloud-to-BC			
3: Gtwy-to-BC			
4: Direct			

Examples

- 1a: IoT Cloud manages all asset data, selectively writes critical data I events to BC
- 1b: Cloud reads other asset' s data for validation, e.g.for pairing of two assets
- 2a: Track & Trace beacons send critical event togateway (e.g. via BLE), gateway forwards event to IoT Cloud, data is evaluated and stored in BC
- 4a: On-asset intelligence is executing BC logic and communication directly

Figure 2. Blockchain & IoT Integration Patterns

1) Asset → IoT Cloud → Blockchain

This integration pattern targets IoT devices without power constraints (e.g., single board computers) that have Wi-Fi or cellular connectivity to communicate directly with the IoT Cloud using IoT data protocols (e.g., MQTT). In this pattern, the IoT Cloud manages all of the asset data and the blockchain serves as the data integrity layer and intra-organizational data plane across multiple Clouds. Finally, the IoT Cloud chooses which data and events are transmitted to and stored on the blockchain.

2) Asset → Gateway/Fog → IoT Cloud → Blockchain

This integration pattern covers resource-constrained IoT devices (e.g., sensors, RFIDs, smart meters) that can only connect to an IoT gateway via low-power wireless communication protocols (e.g., ZigBee, Z-Wave, LoRa). The asset passes data to the gateway, which then processes and forwards the data to the IoT Cloud. The blockchain plays a similar role as described in Integration Pattern 1.

3) Asset → Gateway/Fog → Blockchain

This integration pattern covers emerging computing paradigms (e.g., edge and fog computing), where IoT gateways and edge/fog nodes directly manage connectivity, storage, processing, and analysis in a distributed fashion. In this pattern, the blockchain replaces the centralized IoT Cloud for controlling and managing IoT devices, gateways, and nodes to realize various asset-related core functionalities.

4) Asset → Blockchain

This integration pattern targets machine-to-machine communication and payment scenarios, where IoT devices equipped with Wi-Fi or cellular modules can run either a light client or full node to communicate with other nodes in a decentralized manner. Smart contracts are used to define the interaction rules between IoT devices, which monitor specific events on the blockchain and take actions accordingly.

Trusted Asset Lifecycle

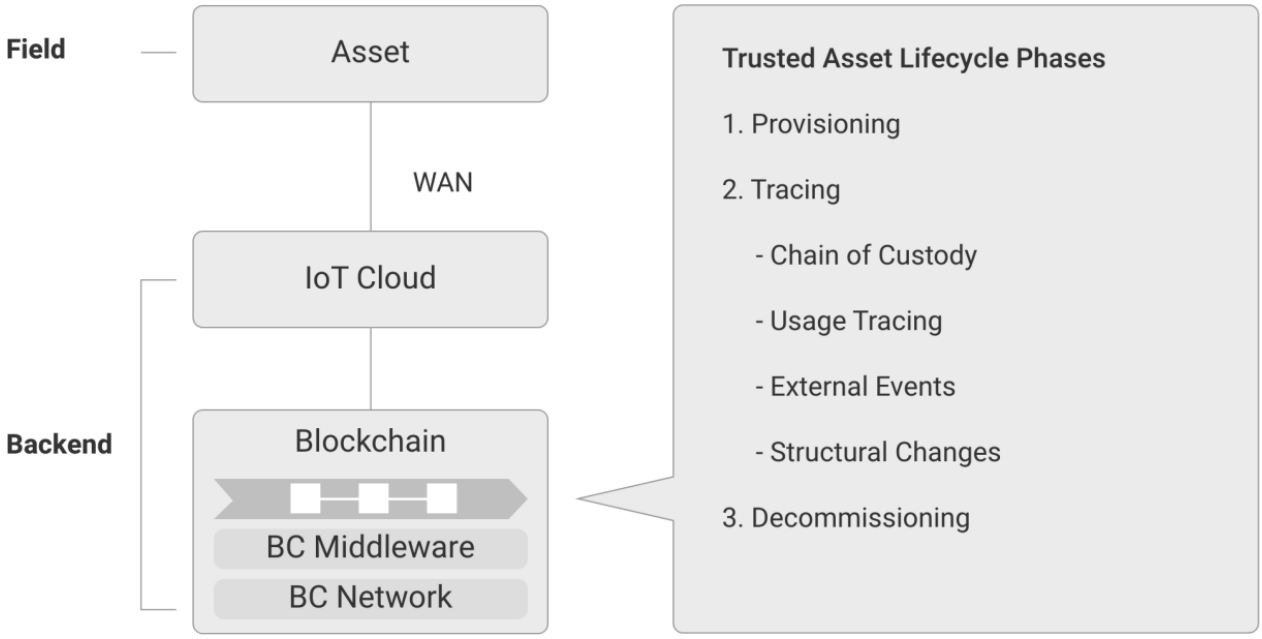


Figure 3. Three-Phase Trusted Asset Lifecycle

Phase I: Provisioning

In this phase, IoT devices are provisioned in a secure manufacturing environment via cryptographic keys and digital certificates. These cryptographic materials are kept in tamper-resistant storage and only accessible by the device's on-board security engine and trusted execution environment (TEE). The provisioned IoT devices are then associated with physical assets and deployed in the field. Once these devices complete authentication with the IoT Cloud, the devices are then registered to the blockchain via smart contracts, thereby enabling blockchain-based device management.

Phase II: Tracing

In this phase, the IoT device continuously reports data (i.e., physical properties of associated assets) to the IoT Cloud, which selectively transmits data "trigger" events to the blockchain. By tracing the history of transactions on the blockchain, one can determine the state transitions of physical assets, including but not limited to the chain of custody, usage frequency, and structural changes. Then by analyzing the state transitions on the blockchain, one can gain valuable insight into the physical assets.

Phase III: Decommissioning

In this phase, IoT devices may be decommissioned from existing linked IoT applications for a variety of reasons (e.g., replacement). In this case, the decommission event will be recorded on the blockchain and the decommissioned devices will be marked as "inactive" on the device registry, which enables any linked IoT applications to adapt to this change accordingly. Throughout this three-phase trusted asset lifecycle, the blockchain keeps track of the state transitions of IoT devices and their associated physical assets, thereby offering an authenticated audit trail for the lifecycle of the physical assets.

Smart Contract Integration Patterns

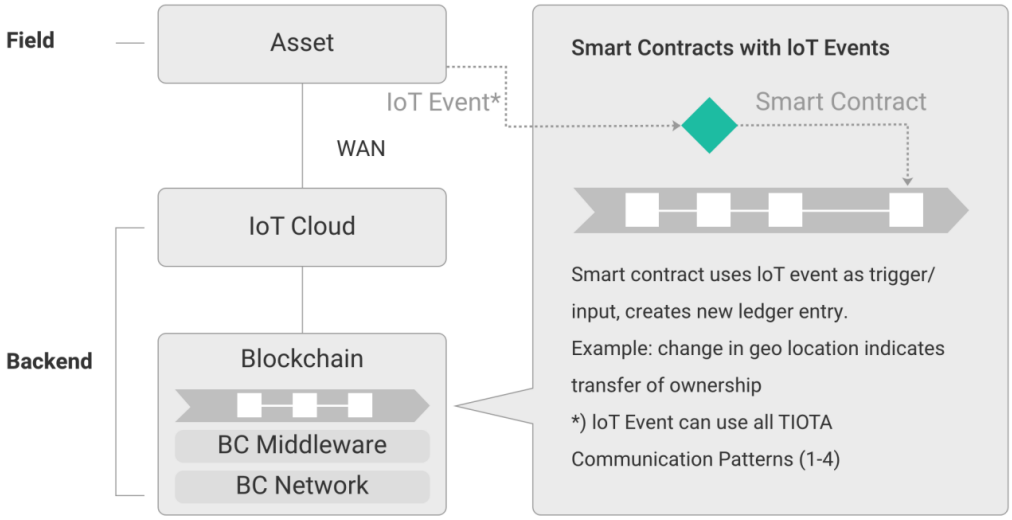


Figure 4. Event-Triggered Smart Contract Integration Patterns

Smart contracts play an important role in the overall reference architecture to enable event-triggered decentralized workflows and cross-chain communication. For event-triggered smart contracts, critical IoT events (e.g., change in geo-location, metric threshold breach) trigger the execution of a smart contract on the blockchain, which facilitates the state change of the IoT device and may incur further actions on the device side. These critical IoT events may be injected into the blockchain by leveraging one of the four Blockchain & IoT integration patterns defined in the previous section.

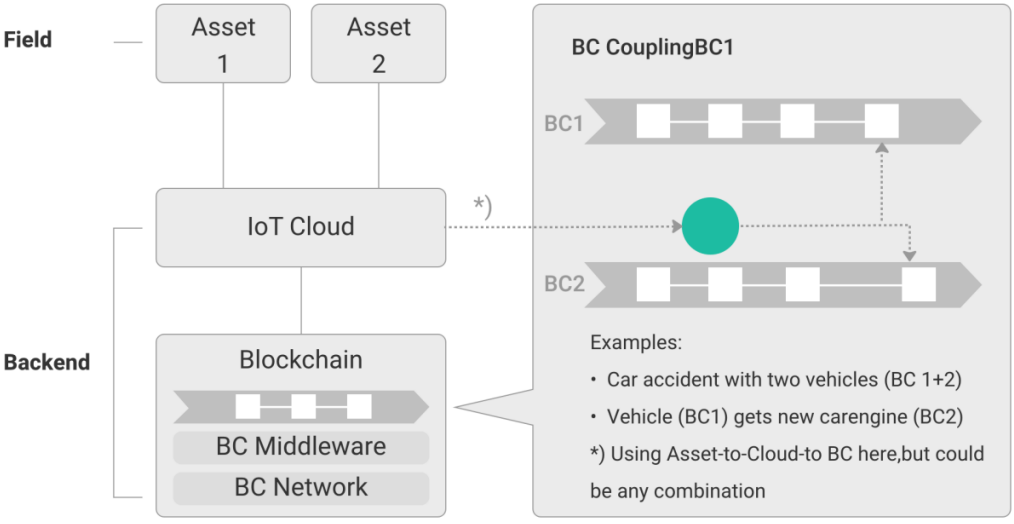


Figure 5. Cross-Chain Communication Smart Contract Integration

For cross-chain communication, the IoT Cloud, which runs either light clients or full nodes for two (or more) blockchains, is able to monitor transactions on both blockchains. Smart contracts deployed on one blockchain can be triggered to execute smart contracts on another blockchain, which is enforced by the bridging node run on the IoT Cloud.