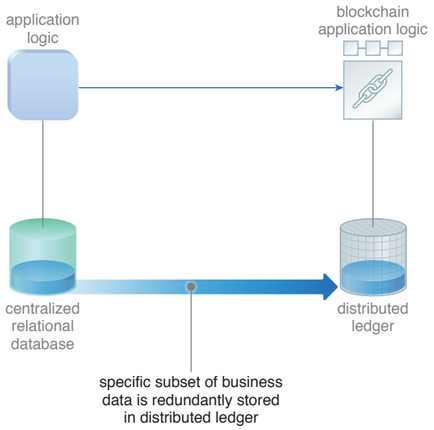
Partial Business Data Capture

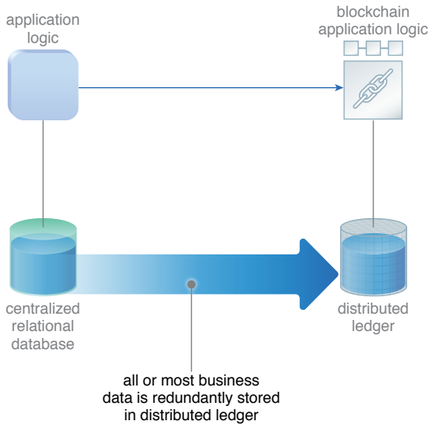
A subset of data residing in a relational database (or another form of legacy repository) can be redundantly stored in a distributed ledger. This model may be suitable when some of the data generated by an application with a centralized database is considered sensitive or high value and for which long-term integrity needs to be preserved. In this case, the distributed ledger can be positioned as an immutable data store responsible for locking the data, as first captured, by ensuring that it can no longer be modified.



A portion of the data in a centralized relational database is redundantly captured in a distributed ledger.

Full Business Data Capture

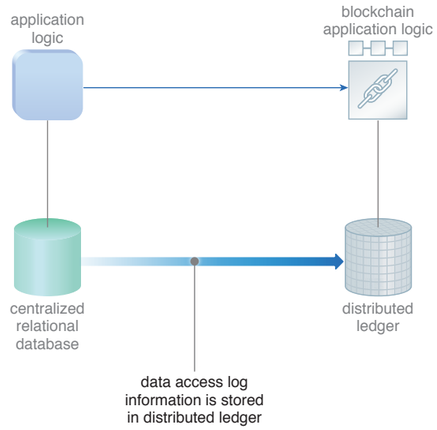
Most or all of the data in a relational database can be redundantly stored in a distributed ledger. This model is not common, but may be considered if the entire contents of a relational database need to be immutably stored and secured. The complexity arising from this model is the import of data stored in relational tables to the distributed ledger's block-based storage system.



A majority of the data in a centralized database is redundantly captured in a distributed ledger.

Log Data Access Capture

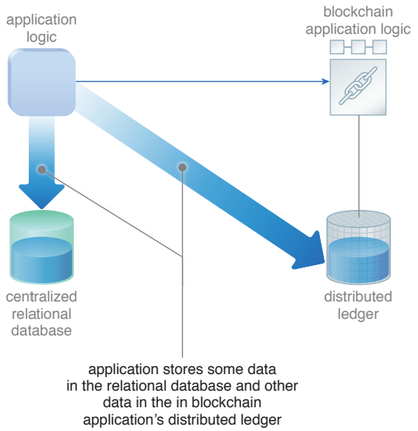
The data access log information of a centralized relational database can be captured in the distributed ledger. This model may be suitable for enhancing the security control over a business database so that if there is ever any question about unauthorized data access attempts or data manipulation that may have occurred, the immutable log repository provided by the distributed ledger can be checked as a reliable "source of truth."



The historical data access log records for a centralized relational database are stored in the distributed ledger.

Partial Business Data Store

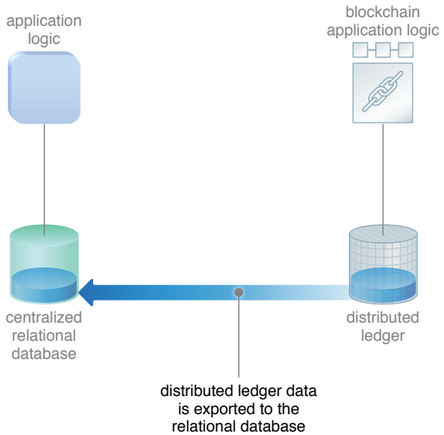
A business application stores a portion of its data in a centralized relational database and other data in a distributed ledger. This model aims to avoid unnecessary redundancy of data across the relational database and the distributed ledger. Records with high performance CRUD (create, read, update, delete) data access requirements are stored in the centralized relational database. Records that require formal validation and permanent and immutable storage are placed into the distributed ledger. This model can still be augmented to allow for subsets of either repository to be redundantly stored.



A business application is integrated with a blockchain application as part of the same overall automation solution. At runtime, the application logic determines whether records should be stored in the centralized relational database or submitted to the blockchain application logic for storage in the distributed ledger.

Ledger Export

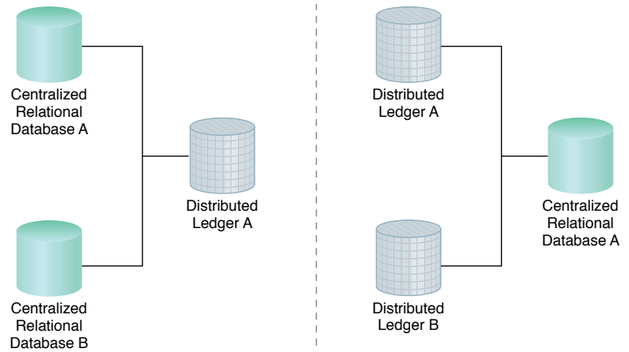
Data records residing in the distributed ledger are exported to a centralized relational database. This model may be suitable for when the data collected by a blockchain application needs to be made available in a more data access-friendly manner. Depending on the nature of the data and the functional requirements of those accessing the data via the relational database, this model may introduce complexity with how records should be structured in relational database tables.



Records in a distributed ledger are exported to redundantly reside in a centralized relational database.

Other Models

So far, the models explored have shown a single centralized relational database alongside a single distributed ledger. There are, of course, no limitations as to how many repositories may be involved as part of a single solution architecture or as part of an integration architecture involving multiple solutions.



For example, a single distributed ledger may be integrated with multiple relational databases or vice versa.

Blockchain Module 1: Fundamental Blockchain
Official Supplement: Distributed Ledger Co-Existence Models

Arcitura® CERTIFIED *Blockchain Architect*

Arcitura® TRAINING Blockchain Architecture

Arcitura® CERTIFIED Blockchain Architect