Autonomous Interdependent Repositories
White paper

Version 1.0

October 15th, 2023

Artem Shamsutdinov

**Abstract**

Autonomous Interdependent Repositories or AIR solves the problem of keeping meaningful data in users hands. It allows separating data from Applications by placing the physical database and the Applications that use it directly on user devices. The Applications and User Interfaces use the data by making queries and running persistence operations in the database. The AIR software maintains retrieval, maintenance and storage of repositories as well as access to the database by Applications.

AIR lets users access the same data via any number of Applications and User Interfaces. It also enables cooperation between Applications. A given Application can provide functionality while other Apps can build logic and data structures that utilize the logic already written in the original App. AIR allows each Application to store data in its own schema. Applications can then reference schemas of other Applications thus promoting cooperative storage across schemas of multiple Applications.

**Previous Work**

AIR works on top of relational databases. Relational databases are the golden standard of databases with all other database types trying to provide the same level of functionality (while increasing performance characteristics or data complexity). Relational databases are easy to understand and are the most widely used database type on the planet.

To accomplish infinite scalability AIR uses composite primary and foreign keys. This is a technique that is allowed by relational databases though is usually considered to be a denormalization. AIR standardized the composite key format which can be logically treated as a single key.

**Core Problem**

The problem that motivated creation of AIR was - how to allow developers write apps on top of relational databases while letting users keep their data and share it with other users. It was specifically designed to make sure that Apps do not store data on servers they own and data is in the direct custody of the users. During the development of AIR a second goal appeared - allowing Applications to cooperate with each other and to allow software developers reuse existing functionality and write only the code that is needed for the features specific to their Application.

Imagine a situation where there are 3 users - Alice, Bob and Charlie. Alice wants to separately share information with Bob and Charlie - to have separate groups of information with each of Bob and Charlie. In turn Bob and Charlie are interested in that and are also interested in sharing some data between themselves but not with Alice. AIR Repositories allow for that to happen:



**How it works**

AIR is a system of splitting relational data into virtual Repositories. A Repository is a collection of rows across any number of tables and schemas. Repositories reside in standard relational databases and any number of Repositories can exist in a given database, in the same exact tables as other Repositories:



Repositories use the following scheme to uniquely identify records:

REPOSITORY\_ID - globally unique repository identifier

REPOSITORY\_ACTOR\_ID - a composite id for a given User using a particular Application

ACTOR\_RECORD\_ID - unique id of a record in a given table in a given database (a database sequence)

Repositories can depend on each other: Interdependent. A given Repository can reference data in another Repository via relational foreign keys. Each Repository maintains copies of original records referenced from other Repositories and can be used on its own: Autonomous. Queries are made across Repositories and transactions can take place across them.

Repositories are shared independently of each other across multiple databases. All involved databases can have different compositions of Repositories at any given point in time. Multiple users can participate in multiple Repositories without having to participate in all Repositories. AIR enables networks of databases where each database contains all of the Repositories a given user needs and none of the Repositories they don’t need:



Each Repository maintains its own Transaction Log and is shared by sharing transaction entries in its. Conflicts in the Transaction Log are resolved by using synchronization timestamps of the transaction entries.

**Conclusion**

AIR enables Fair Data Ownership and Independent Software Development. Fair Data Ownership is attained by keeping meaningful (relational) data in users hands. In AIR independent Applications cooperate with each other.

You can find more information at:
<https://beyond-decentralized.world/documentation.html>

And follow the progress of AIR implementation at:
<https://github.com/beyond-decentralized>