THE R* PROJECT:

The R* project is developed my IBM. The goal of R* project is to build a distribution database made of co-operating but autonomous sites at each one supporting a relational database system.

The R* is a natural extension toward distributed environment of system R, a relational database management system prototype, which is available as IBM product.

In R* data are stored as relations. It does not support fragmentation and replication. Different R* modules can be placed on the same computed for security accounting or performance reasons.

The most important aspect of R* is to provide site autonomy. Site autonomy is achieved when both sites can access control data without referring each other.

The location transperancy can be achieved in R* (ie) the user is not aware of the location where the data is stored.

ARCHITECTURE OF R*:

R* is composed of three major components

1. Local database management system.

2. A data communication component.

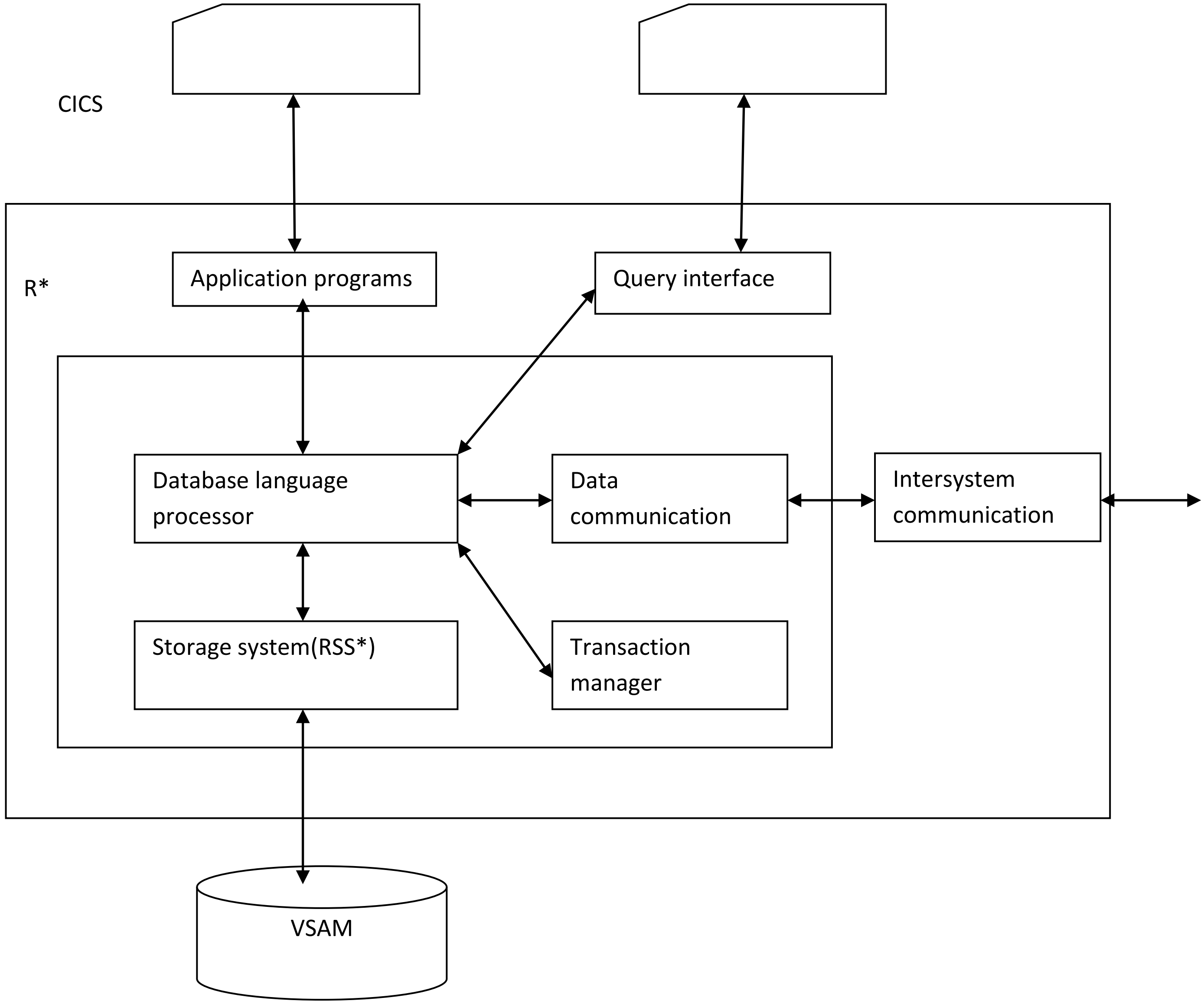
3. A transaction manager

Which co-ordinates the implementation of multisite transactions.

The local database management system is divided in to two components.

1. A storage system: Storage and retrieval of data.

2. Database language processor: Translates high-level SQL statements in to operations on storage system. The storage system used in R* project is called **RSS*.**



Architecture of the R* system

The R* communicate via the Inter System communication(ISC) facility of CICS. Each R* site runs in a CICS address space and CICS address space and CICS handles terminal I/O and message communication.

An application program makes all database access request to the R* system at it's local site. All intersite communications are between R* systems at different sites. Thus in an R* environment there is no need for remote application programs.

The transaction manager at the site can be used to initiate the transaction by implicitly performs a begin and end transaction. After the completion of work done a commit is performed.

R* can also be invoked using a "user-friendly-interface"(UFI). SQL statements are submitted from UFI to R* and each statement is treated as an individual transaction.

To each transaction the transaction manager assigns a unique transaction identifier, made with the local transaction counter and identifier to that site.

R* uses the “process” model of computation. Instead of allocating a different process to each database access request a, a process is created