Software Design and Architecture

Data Centered Architectures and Repository architecture

Data Centered Architecture

- The data centered software architecture is characterized by a centralized data store which is shared by all surrounding software components.
- The software system is decomposed into two major partitions: data store and independent software component or agents.
- The connections between the data module and the software components are implemented either by explicit method invocation or by implicit method invocation.

- There are two categories of data centered architecture: *repository* and *blackboard*.

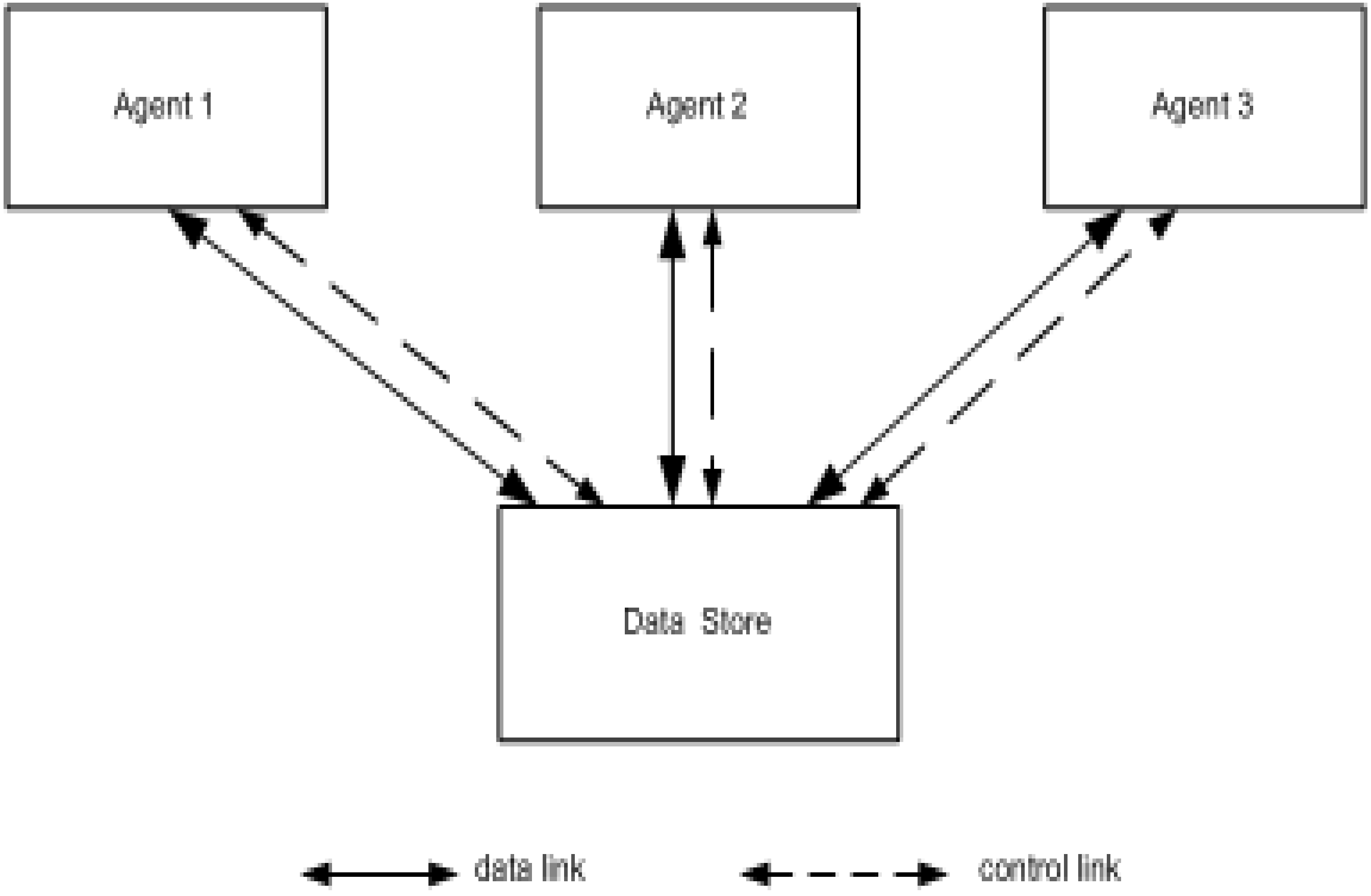
- These are differentiated by the flow control strategy.

- The data store in the repository architecture is passive and clients of the data store are active; that is, clients (software components or agents) control the logic flow.

- Clients may access the repository interactively or by a batch transaction request.

- The data store in the blackboard architecture option is active and its clients are passive; thus, the flow of logic is determined by the current data status in the data store.

- The clients of a blackboard are called knowledge sources, listeners or subscribers.

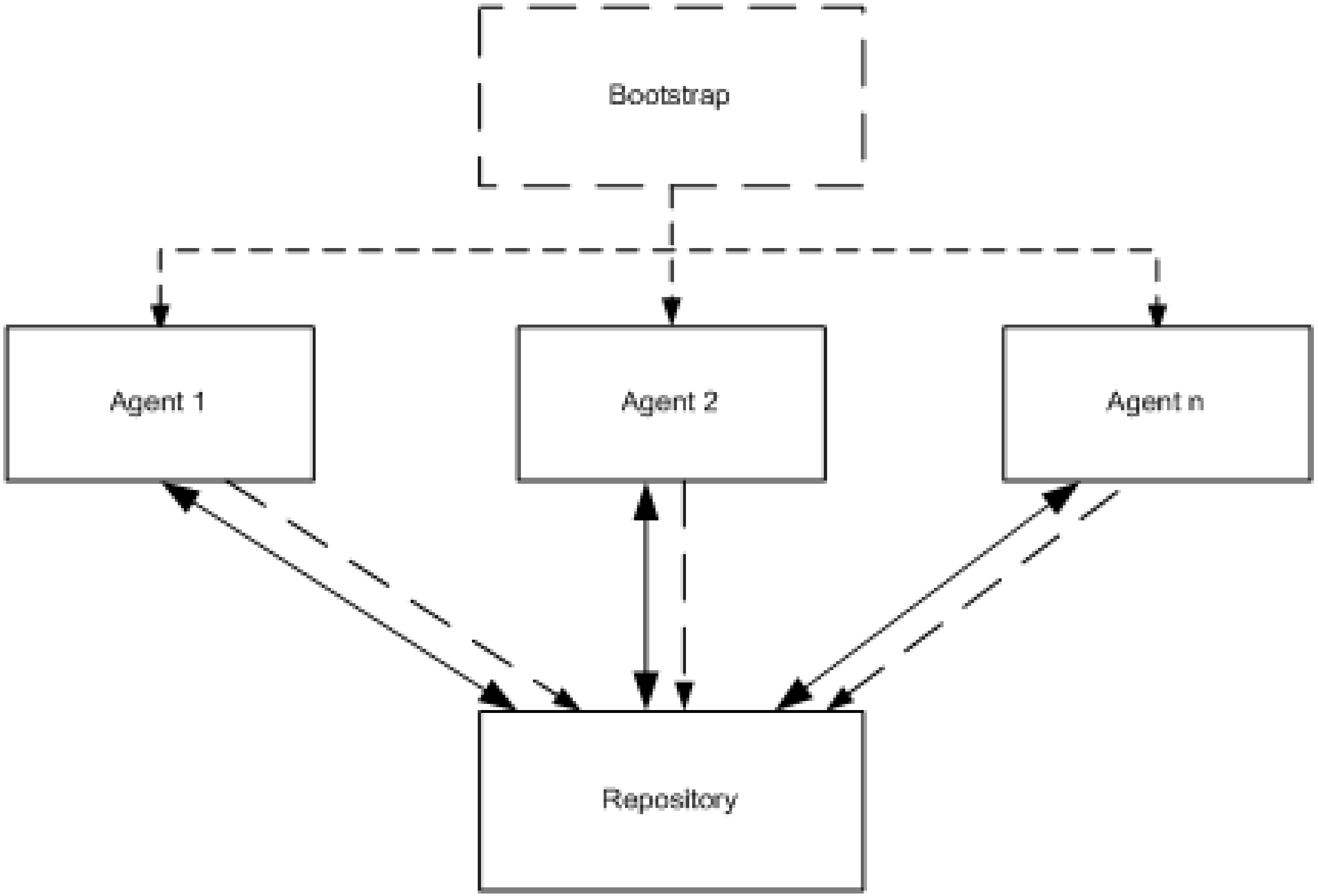


Block diagram of typical data centered architecture

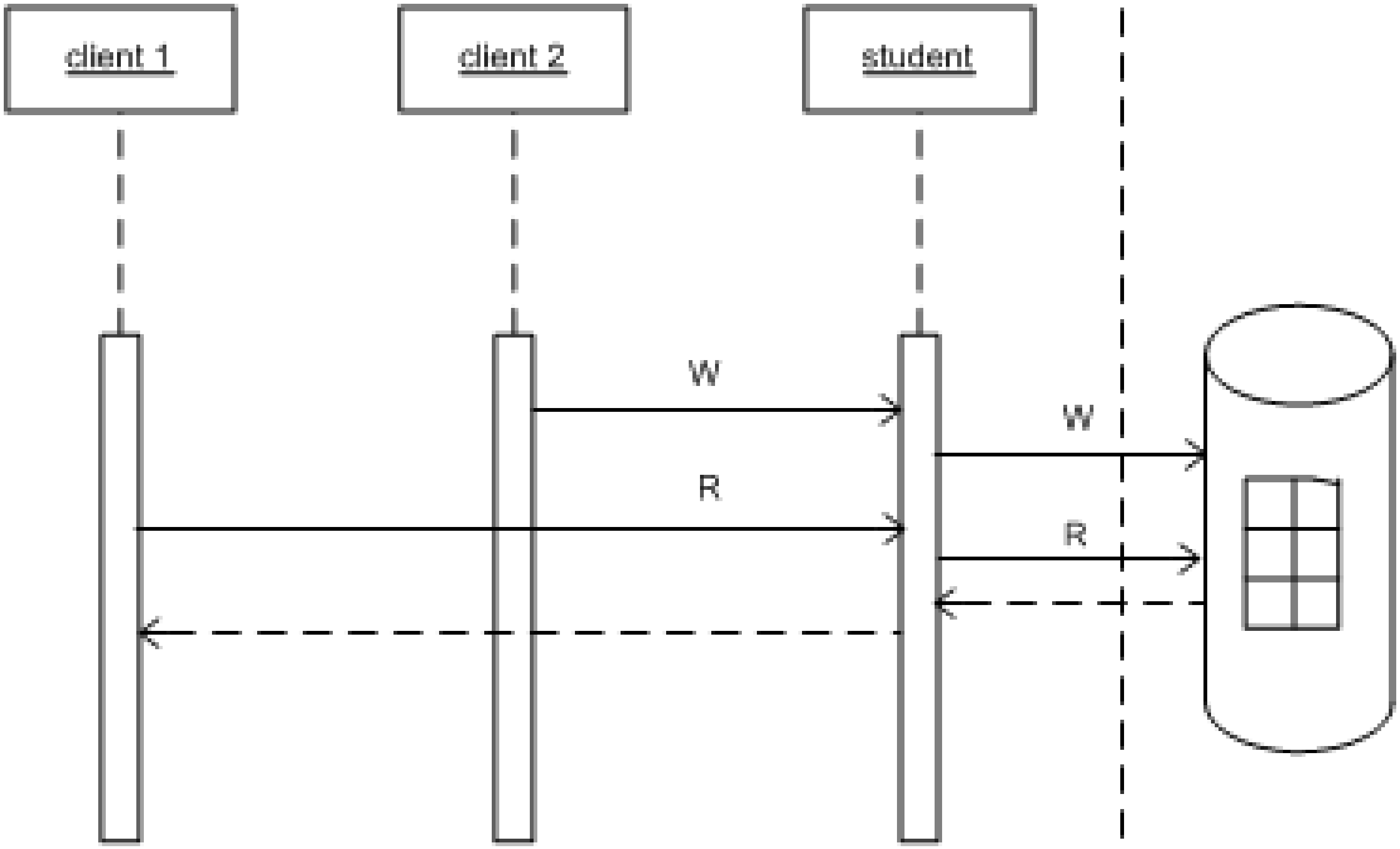
Repository Architecture Style

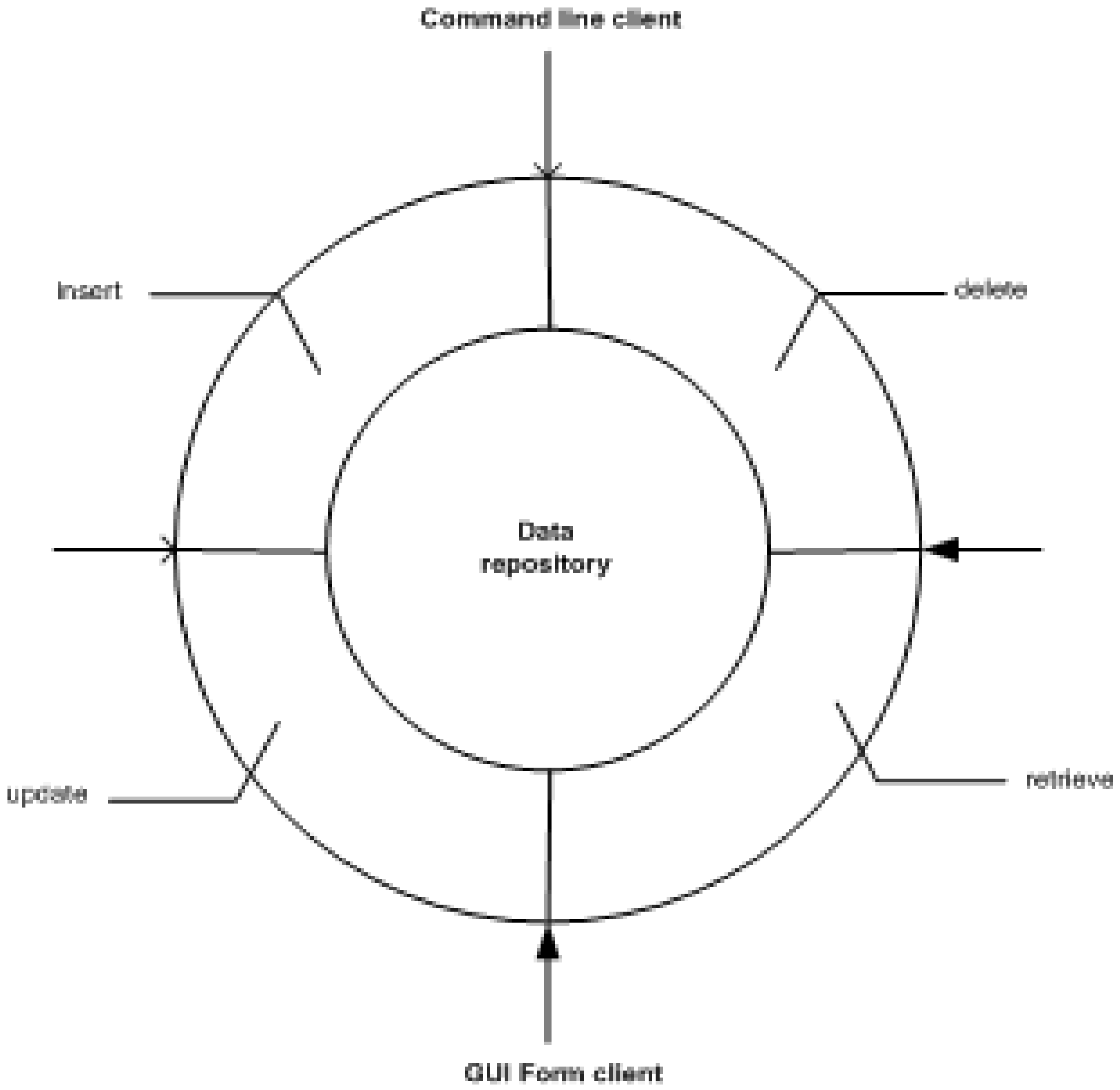
- The repository architecture style is a data centered architecture that supports user interaction for data processing.

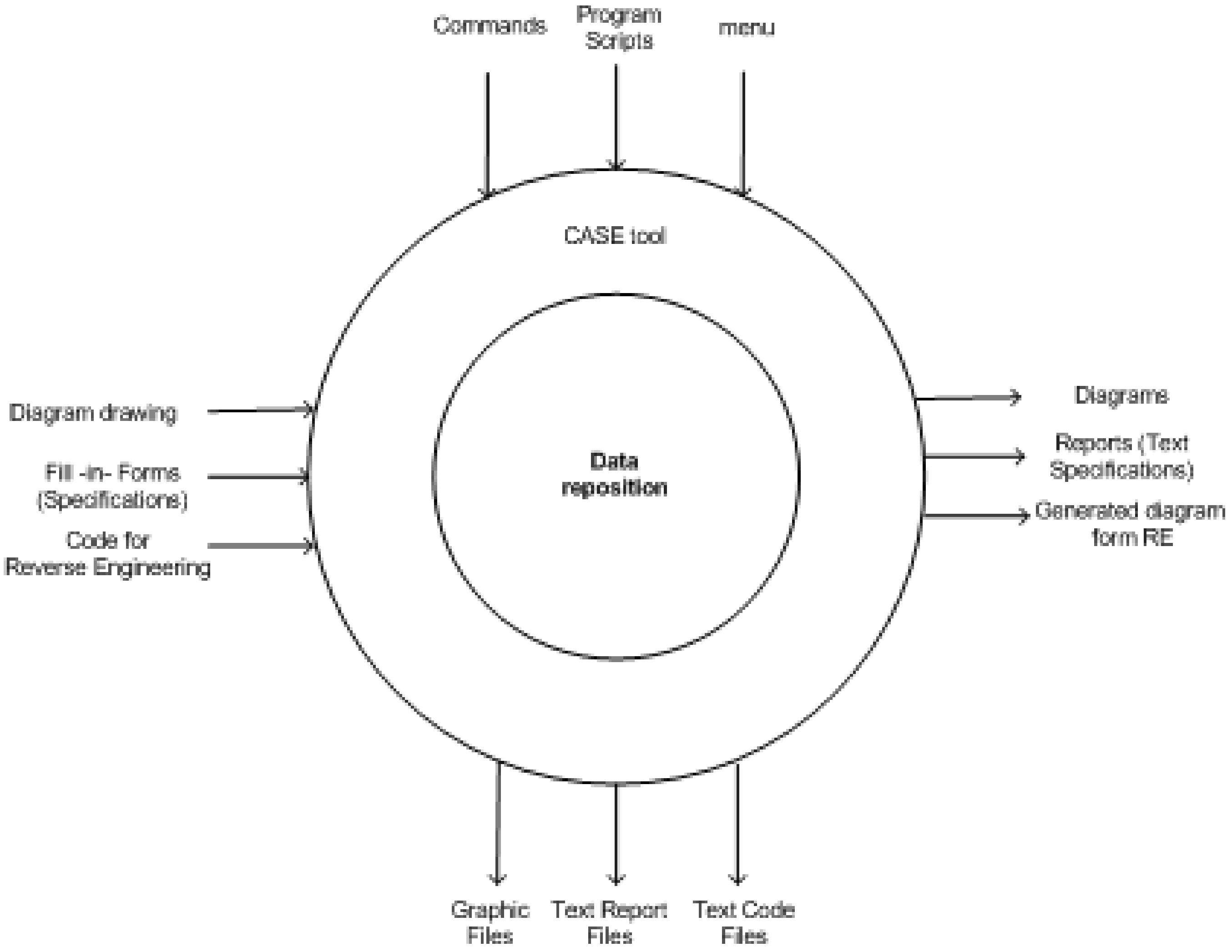
- The software component agents of the data store control the computation and flow of logic of the system.

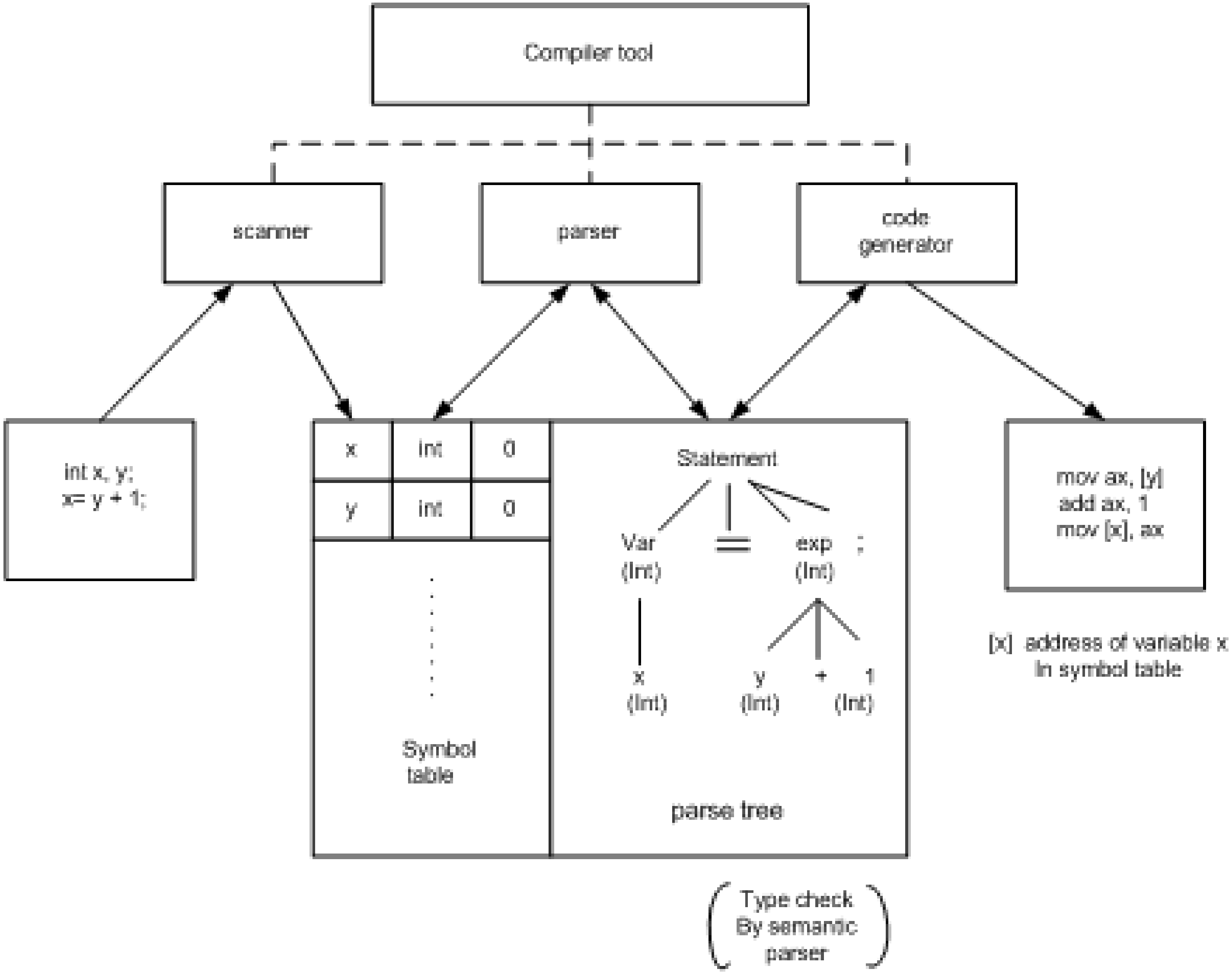


Block diagram of repository architecture


client 1
client 2
student
W
W
R
R


Command line client
insert
delete
Data
repository
update
retrieve
GUI Form client


Commands
Program Scripts
menu
CASE tool
Data reposition
Diagram drawing
Fill -in- Forms (Specifications)
Code for Reverse Engineering
Diagrams
Reports (Text Specifications)
Generated diagram form RE
Graphic Files
Text Report Files
Text Code Files



```
int x, y;
x= y + 1;
```

x	int	0
y	int	0

```
mov ax, [y]
add ax, 1
mov [x], ax
```

Applicable Domain of Repository Architecture

- Suitable for large complex information systems where many software component clients need to access it in different ways.

- Data transactions to drive the control flow of computation.

Benefits

- Data integrity: easy to backup and restore.
- System scalability and reusability of agents: easy to add new software components because they do not have direct communication with each other.
- Reduce the overhead of transient data between software components.

Limitations

- Data store reliability and availability is a very important issue. Centralized repository is vulnerable to failure compared to distributed repository with data replication.
- High dependency between data structure of data store and its agents.
- The changes of data structure have significant impacts on its agents. The data evolution is more difficult and expensive.
- Overhead cost of moving data on network if data is distributed.

Summary

- Introduction to Data Centered and repository Architecture.
- Benefits of repository architecture.
- Limitations of repository architecture.