# Describing of the project

This is a student project created during the course “Software engineering and cloud computing”. It is a monitoring system of the bitcoin network, using it you can trace bitcoins’ transactions and blocks. Also you can construct different charts and diagrams.

This project consists of Bitcoin client, Bitcoin worker role, storage implementation, storage worker role and webrole. This project is created for Windows Azure virtual machine. It uses Windows Azure Storage and other cloud technologies of Windows Corporation.

This application is written with C#.

## Schema of the application



## Bitcoin client

For our application we use standard bitcoin client <HERE NAME OF THE CLIENT>. It is launched on the Windows Azure Virtual Machine (VM). We are taking new transactions and information about blocks from it.

## Bitcoin Worker Role

*BitcoinWorkerRole.csproj*

This part takes data from the Bitcoin Client. It also converts received data to the XML format and extracts transactions from the blocks. It uses Storage implementation for uploading data to the Azure Storage. Blocks are saved to the BocksContainer and Transactions are saved to the Transactions container. Also this part adds messages to the queue about which block should be processed next.

## Storage implementation

*Storage.csproject*

This is a service part of the project for communicating with Windows Azure Storage. It contains with:

* Serilization – saving different data models to the storage in XML formatl;
* Models – in this folder different structures of data are described. All these models are serialized;
* Coding /not implemented/ – zipping and unzipping data to minimize data exchange;
* Managing Windows Azure Storage – staring and connecting to different containers.

We use three containers and a queue of Windows Azure Storage.

In the BlockContainer we save information of the blocks like a XML file. We use block’s hash to name each file. In these files all information about bitcoin blocks are saved. To minimize file’s size we removed transactions and added just their hashes.

In the TransactionsContainer we saved transactions XML by their hashes. Also we added hash of the block which is owned this transaction to the XML structure.

In the HigherContainer we save transactions’ hashes by time to accelerate data processing and access to the data from the UI.

## Storage Worker Role

*StorageWorkerRole.csproj*

In this project we have made data processing, adding references to the higher container, collecting statistical information and saving it to the storage.

## Web Role (also a user interface)

*WebRole.csproj*

A web site with a user interface (UI).

* Has a controller, which is used for taking data from the blob storage;
* View – for creating and grouping data which was demanded. (Adding data to the table, creating data for graphic construction)