**3.6 Global software control**

The CrypDist system architecture is to have a decentralized, explicit software control. To respond users’ quality demands and to minimize the problems followed by the distributed nature of the system, CrypDist base its content, processing, and collaboration distribution activities on the event control flow paradigm.

To process the composite events in the system such as upload of modified raw data or query data, careful placement of event-services must be done to with minimum resource usage and minimum delay based on replication and load balancing techniques [1]. Since CrypDist consists of interconnected nodes with similar tasks and functions to execute as in peer-to-peer system, exchange of events between pairs is provided. Beside the event-driven control, the content stored in the blockchain will be updated dynamically on panels on the main screen of the user.

**3.7 Boundary conditions**

* **Start-up:** The system is started by initializing blockchain services. To interact with user, system provides a main screen to user.
* **Shut-down:** The system will only be terminated in case of maintenance or update of servers. User interface and services on local systems will be terminated when user chooses to.
* **Exception handling:**
* **A hardware failure:** One or more server that store raw data may fail. In case of such a failure, the system automatically detects and suspends problematic data centers or servers.
* **Network outage:** In case of a network outages between pairs, followings are the cases:
  + The blockchain synchronization on local servers may fail. If any peer may upload a new file or update file, synchronization among peers may not be provided in a short time. The system detects the situation and protects the current state.
  + Replication of raw data among distributed servers may fail. The system detects and protects the current state.
* **Power outage:** Power outages may occur in one or more server. In case of such a case, the system operations continue via other servers. And nonfunctional server will be restarted after power failure.
* **Data corruption:** Due to inconsistencies in transmission, data might be corrupted and corrupted data might be uploaded to system. In case of such situation via replication of data on distributed servers and version control service, the system may recover the data.
* **Monitor blockchain content:** The blockchain content on the main screen cannot be uploaded automatically.
* **Data Entry:** The system fails when the user is entering information.
* **Logging out:** The user is unable to logout.
* **Logging in:**

**User errors:** Displays an appropriate error message to user if any below case occurs.

* Username is not matched.
* Password is not matched.
* Length of the input to username text field is lower than the allowed length.
* Length of the input to password text field is lower than the allowed length.
* Username text field contains unrecognized characters.
* Password text field contains unrecognized characters.

**System errors:**

* The authenticated main screen is not displayed after logging in.

**[1]** . Xhafa, F., Paniagua, C., Barolli, L., Caball´e, S.: A Parallel Grid-based Implementation for Real Time Processing of Event Log Data in Collaborative Applications. Int. J. Web and Grid Services, IJWGS 6(2) (2010) (in press)