High Level OEC Synchronizer System Design

Data Fetcher
- Download CSV files from monitored sites
- Parse CSV files content into separate rows for further action

Analyzer
- Compare the updated value from the NASA and Exoplanet and find the latest update of the two

Git Action
- Create new branch
- Commits the updates/newly items
- Create a pull request to notify the user
- Make sure the pull request are created on system by system basis

Update checker uses the output of data fetcher to get the last updated field

Passes data for xml file update

Analyzer secures the updated value and then passes it to the XML Creator/Updater

Clone Forked OEC Repo & Update Checker
- Clones the forked OEC Repo to get existing files in the OEC
- Uses the last updated time field to check if item was updated
- Finds OEC equivalent of the updated item
- If equivalent item is not found then takes the data as newly added item

Passes data for xml file creation

XML Creator/Updater
- Ensure the the value have same SI unit as OEC
- If not then carry out conversion
- Ensure name alias and Human error is handle
- Updated values are merged into the existing xml files
- New xml file is created for new planet

Git Action gets the changes to commit from XML Creator/Updater
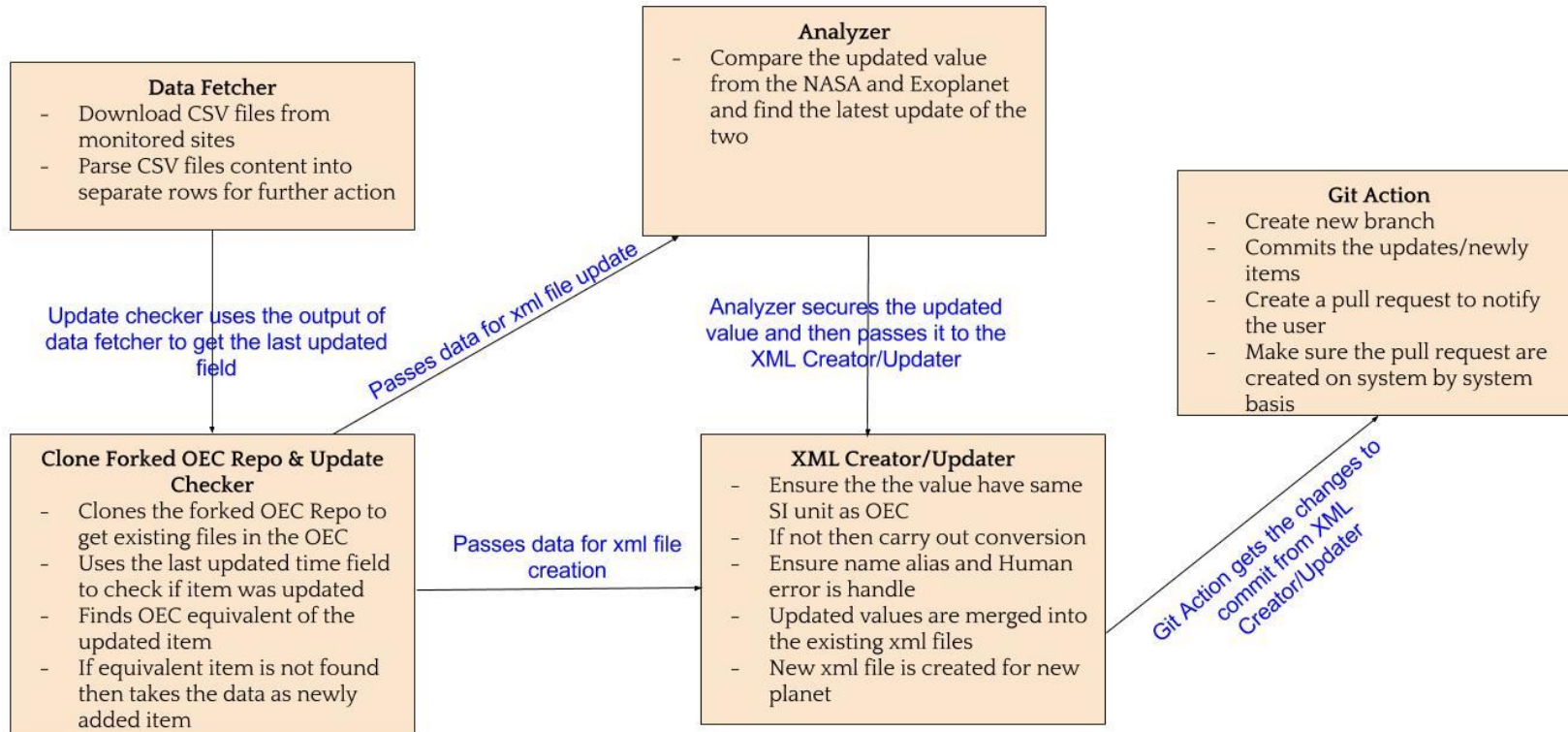
The system design had minor change from deliverable 3 to deliverable 4. The software is now designed to make a clone to a forked OEC repository every time to get the latest version of the xml files. Update checker finds the OEC equivalent of the updated item. In the system designed from deliverable 3, xml files are created every time. In deliverable 4, we have changed this part of the design where new xml files are created if the planet does not currently exist in the OEC. If the updated item is from an existing planet, the analyzer compares the two updated values from the monitored catalogues (NASA and Exoplanet) and chooses the value from the catalogue that was updated most recently. This updated value is then passed to the XML Creator/Updater where the updated value is merged to the existing xml file. Lastly, the changes are committed and pull requests are sent to the client.