PRICE REQUEST MODE (API SUB. ON)

BROWSER

1, 5, 8, 8.8

2, 4.1, 6.1, 6.4, 8.1, 8.4, 9

WOOCOMMERCE SITE + 3DPRINT PLUGIN

3, 4.3, 6.6, 8.5, 11, 12, 13
15, 16

4.2, 6.2, 6.5, 8.2, 8.5

API SERVER

6.3, 8.3

4.4, 6.7, 8.7

4.2, 6.5, 8.5

8.6
10
14
17

15

12

DATABASE
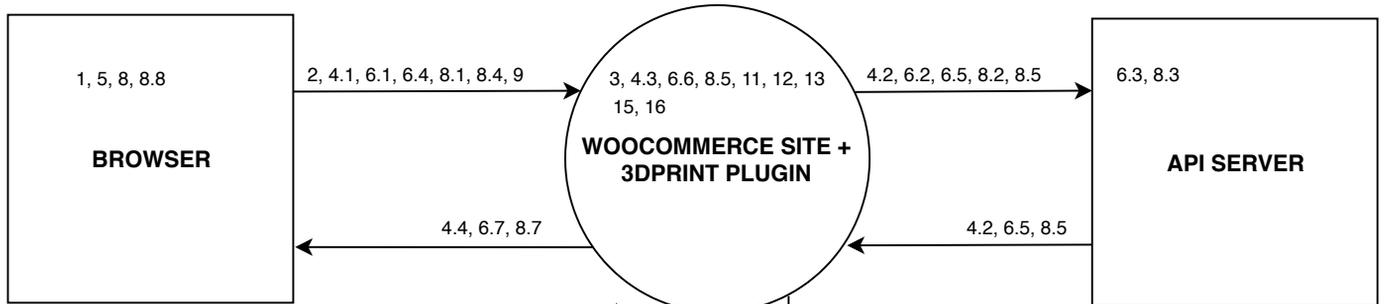
1) A model file is selected in the browser.
2) The file is sent to the site by chunks.
3) All file chunks are uploaded and joined on the woocommerce site.
4) If a file conversion is required (non-stl, non-obj) then:
 4.1) The browser sends an ajax post request to the woocommerce site.
 4.2) The plugin posts the file to the API server synchronously and receives the download link in one shot.
 4.3) The converted file is downloaded and stored in the 3dprint upload folder
 4.4) Browser receives the ajax post response with the download URL and displays the new model in the browser.

5) By this time the model should be displayed in the 3D viewer. On the browser side some JavaScript calculations are done:
dimensions, bounding box, surface area, material volume (100% solid).

6) If a file repair is enabled:
 6.1) The browser sends an ajax post request to the woocommerce site.
 6.2) The plugin posts the file to the API server.
 6.3) The API server runs the repair job.
 6.4) Every 3 seconds the browser initiates an ajax post request to the site checking the repair job status.
 6.5) The plugin sends a repair check request to the API server.
 6.6) Once the repair is finished the repaired file is downloaded and stored in the 3dprint upload folder.
 6.7) Browser receives the ajax post response with the download URL and displays the new model in the browser.
7) File optimization logic works the same way as the file repair.

8) If API subscription is enabled the browser initiates the model analysis process.
 8.1) The browser sends an ajax post request with the user chosen parameters (printer, material, coating, infill, unit) to the woocommerce site.
 8.2) The plugin posts the file with slicing parameters to the API server.
 8.3) The API server runs the model analysis job.
 8.4) Every 3 seconds the browser initiates an ajax post request to the site checking the analysis job status.
 8.5) The plugin sends an analysis check request to the API server.
 8.6) Once the analysis is finished the plugin receives the data from the API server (material volume, print time) and stores it in the database.
 8.7) Browser receives the ajax post response with the material volume and print time and displays them on the screen.
 8.8) Browser calculates the estimated price and displays it on the screen.
9) If the "request a quote" button is clicked the browser posts the price request form along with the estimated price.
10) The plugin updates the price request table with the form values.
11) The plugin sends a notification e-mail to the admin configured address.
12) The admin user goes to the price requests page and sees the list of current price request.
13) The admin selects a price request, sets the price/comments and clicks "E-mail a Quote"
14) The price request status changes to "Quote sent"
15) The customer clicks the "buy now" link received either through the the e-mail or the "my account" page.
16) The product with the admin set price is added to the shopping cart and the standard WooCommerce checkout comes into play.
17) WooCommerce order saved in database