Data Centered "Blackboard" Architectural Style

Rationale:
I have chosen to use the data centered blackboard style. This will consist of a central data data repository and other software components monitoring and communicating through it. Implementing the information system using the data centered black board style is suitable for the given problem for the following reasons:

- Using a central location for all data storage allows for easy up-scaling. The same structure could be used for multiple pumps and cashiers.
- Using implicit invocation allow as simplistic and modular approach to the design. If a component want another component to perform a task it will simply ask for "someone" to perform that task via the data repository / blackboard.
- As the data is all in one place it would be easy to implement another component that does some analysis on the data for marketing / finance purposes.

Component and connector description:

1 **Pump Input**
1.1 These are the raw inputs coming from the real world. These include the pump pick up sensors to detect if a pump has been picked up and which one, and the fuel flow sensors.

2 Input conditioner
2.1 This is an active software component that looks at the raw information that has been placed in the data repository by the input sensors and converts the inputs into something meaningful. It will provide the following functions:
2.1.1 It will take the raw flow sensor input (12bit float representing flow rate), convert it into liters per second and be responsible for performing the integration required to get the volume of fuel taken / being taken in real time.
2.1.2 Calculating the cost of the fuel being taken using the volume as calculated from the flow sensors, the type of fuel being taken inferred by the nozzle pick up sensor and the price per liter for a given fuel as stored in the data repository.

3 Pump user interface
3.1 This module will take the conditioned inputs from the data repository (as conditioned by the input conditioner module) and provide a user display at the pump. This is a passive component and will be triggered by the input conditioner module (nozzle pickup). It will display the current amount of fuel taken and the current price for that volume of fuel.

4 Transaction Manager
4.1 This is an active software component. It will perform the following functions:
4.1.1 It will continuously monitor the data being accessed and modified by the input conditioner module. Once it detects a volume of fuel has been taken and the nozzle has been put back in the holder it will generate a transaction record in the data repository. This will contain the pump ID, the type of fuel taken, the amount taken and the price of the transaction.

5 Cashier session
5.1 This is a passive software component that becomes active once a cashier has logged in, once the cashier has logged in this will stay active until the cashier has logged out. It will provide the following functionality:
5.1.1 If a transaction is not currently in the control of another instance of the cashier

session it can take control of that transaction in the data repository created by the transaction manager. It will then mark this transaction record as “locked” to stop concurrent access. It will do this by setting a bit in a status word contained within the transaction record.

5.1.2 Request a payment to the “secure payment” module via the data repository. In practical terms this will set a bit in a status word in the transaction record that the “Secure Payment” module will be looking for.

5.1.3 Mark the transaction as being completed or void. Again, this will be done by setting setting some bits in a status word in the transaction record and depends on the outcome of the operations performed by the secure payment module.

6 Cashier user interface

6.1 This is a passive software module that will be triggered by the cashier session. This module is just the user interface and will allow the cashier to manipulate a transaction record.

7 Secure Payment

7.1 This is an active software component that is continuously monitoring the transaction records status word. When it detects a transaction that needs to be paid it will carry out the secure payment. Once completed it will set the status word to “paid” or “void” so the cashier session can complete the transaction and unlock the record.