Title: Transferring Data Between Modules in a Modular Monolith

Date: 2024-01-25

Status: Draft

Problem

We have a modular monolith of a shopping setup. The following modules exist:

- Users
- Shopping
- Books

During the shopping process, we may need to get the book price from the Books module and the user's payment info from the Users modules. How could we get these details from the other modules to the Shopping module?

Options

We asked the team for options on this. These are the proposed options:

- Public API (HTTP)
- Public message-based query
- Just direct method call
- Duplicate the data (materialized view)
- API gateway of all data needed
- Add everything you need via parameters on the call to add it

Context

We're in one solution.

Thoughts & Trade-Offs

- Public API (HTTP)
    - This would work better in microservices.
    - If we move to microservices, this would be taken care of.
    - However, we're in a monolith, so this might not make sense.

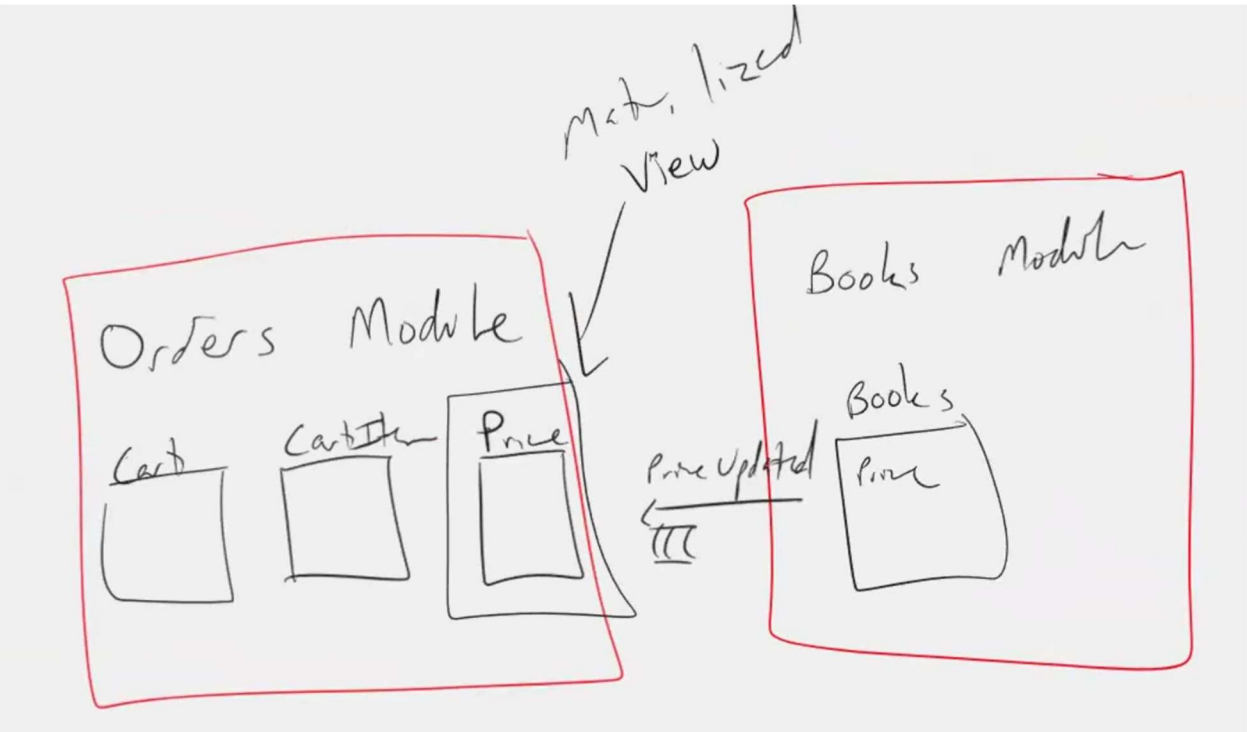
- Public message-based query
    - Put a message on the bus. "Prepare it for me."
    - Could be a big payload
    - Asynchronous messages - outer process could sit and wait for it.
    - We might have to spin and wait over an async process.
        - What if the queues are full?
        - What if things aren't happening instantly?
        - How do we handle timeouts?
        - Do we end up blocking someone from adding something in their cart?
- Just direct method call
    - Something needs public in the other module in order to access it
    - Direct dependencies or reflection
    - Leads to a spider-web of tangledness
- Duplicate the data (materialized view)
    - Preferred - totally decoupled
    - Have a version of the data you need locally
    - Update the local copy in an async manor
    - Usually a better trade-off
- API gateway of all data needed
    - You might expose this at the app level.
    - The modules only need to know about the public API and not the other modules themselves.
    - But this has the same kind of downfalls of the API over HTTP in a monolith.
- Add everything you need via parameters on the call to add it
    - This moves the problem. Where do the people who are calling this get their data?
    - The shopping module still needs to know how to get that data to put it in parameters.

Decision

- Use a local materialized view and update it asynchronously.

Consequences

As we re-evaluate our decision, we drew out an example:



There could be some issues with this.

- Duplicated data - This could get large, depending on how many properties are duplicated.
- Consistency issues