Component based architecture

The idea is to separate bounded contexts (DDD) in components.
Each component will encapsulate the implementation details in an Internal package no accessible from outside (**testable rule**)
and provide APIs for external communication: (port-adapter pattern/events)

1. Output API/Out Ports/events: to be consumed by other Components (implemented in the internal package in the component)
2. Input API/Input Ports: to be provided (implemented) by other Components (they will implement the adapters/consumers in their internal package)

Because we don't have a rich domain, (in my opinion) it's not necessary to implement all the DDD elements inside each component.
The aggregates/entities functionalities/behavior/validation are delegated to component services to simplify and improve readability.
This could be subject to discussion and open to change.