

Auth

Actor - Account (Admin role)
Command - Create Account
Data - Account
Event - Auth.AccountCreated

Task Tracker

Actor - Account
Command - CreateTask
Data - Task + Account public id
Event - Task.TaskCreated

Actor - Account (Admin, Manager)
Command - Assign Tasks
Data - Tasks + Account public id
Event - Task.TaskAssigned

Actor - Account
Command - Close Task
Data - Task
Event - Task.TaskClosed

Actor - Account
Command - Get Account Tasks
Data - Task(s)
Event - Task.TasksFetched (not used)

Accounting

Actor - Auth.AccountCreated event
Command - Create Balance record
Data - Account
Event - Accounting.BalanceCreated

Actor - Task.TaskCreated event
Command - Create prices
Data - Task public id
Event - Accounting.PriceCreated

Actor - Accounting.PriceCreated event
Command - Change balance
Data - Task public id + Balance + Price
Event - Accounting.BalanceChanged

Actor - Task.TaskAssigned event
Command - Change balance
Data - Task public id + Balance + Price
Event - Accounting.BalanceChanged

Actor - Task.TaskClosed event
Command - Change balance
Data - Task public id + Balance + Price
Event - Accounting.BalanceChanged

Actor - EndOfaDayCron
Command - Send email
Data - Account public id + Balance
Event - Accounting.EndOfaDay

Actor - Accounting.EndOfaDay event
Command - Reset Balance
Data - Balance
Event - Accounting.BalanceReseted

Actor - Accounting.BalanceChanged event
Command - Create audit record
Data - Task public id + Balance publi Id + Price
Event - Accounting.AuditLogCreated (CUD)

Actor - Accounting.BalanceReseted event
Command - Create audit record
Data - Balance + Acc
Event - Accounting.AuditLogCreated (CUD)

Actor - Account (regular)
Command - Get Balance info
Data - Balance + Audit
Event - Accounting.BalanceInfoFetched (not used)

Actor - Account (admin + accountant)
Command - Get management balance info
Data - Audit
Event - Accounting.ManagementBalanceInfoFetched (not used)

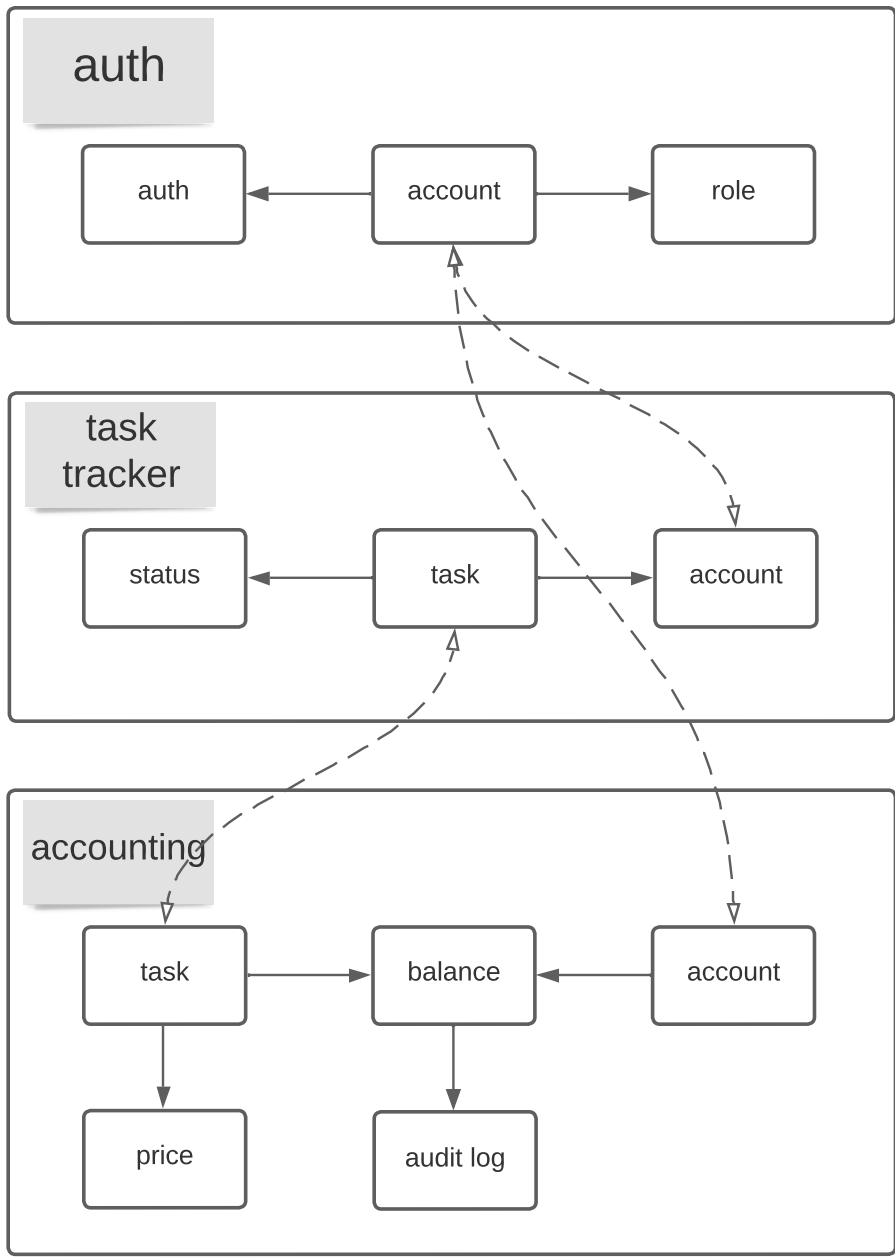
Analytics

Actor - Accounting.BalanceReseted event
Command - Save daily stats and clean data
Data - Audit logs + Tsks + Prices
Event - Analytics.AnalyticsStored (not used)

Actor - Account (admin)
Command - Get analytics info
Data - Audit logs + Tasks
Event - Accounting.AnalyticsFetched (not used)

Я подумал, что можно сделать единственный бизнес ивент это сохранение самой дорогой задачи и заработка менеджмента за сегодня в конце дня (ну и очистка лишних данных, например). А в остальном просто собирать данные из CUD эвентов (или по CUD эвентам?) и по ним уже для пользователей любые запросы строить

С одной стороны можно было бы сделать генерацию цен задачи прямо в таск-трекере, но таск-трекер ведь не продвинут и это как бы другой бизнес процесс. Еще был вариант при создании задачи создавать цену и выдавать событие обратно про создание цены, чтоб в таск-трекере ей привязали к задаче, но это что-то сложное и ненадежное. Решил создать отдельные данные



Actor
Command
Event
Read Model

