

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

Accounting

Actor - Accaunting.TaskCreated event
Command - Change balance
Data - Task public id + Account public id
Event - Accaunting.BalanceChanged

Actor - Task.TaskAssigned event
Command - Change balance
Data - Task public id + Account public id
Event - Accaunting.BalanceChanged

Actor - Task.TaskClosed event
Command - Change balance
Data - Task public id + Account public id
Event - Accaunting.BalanceChanged

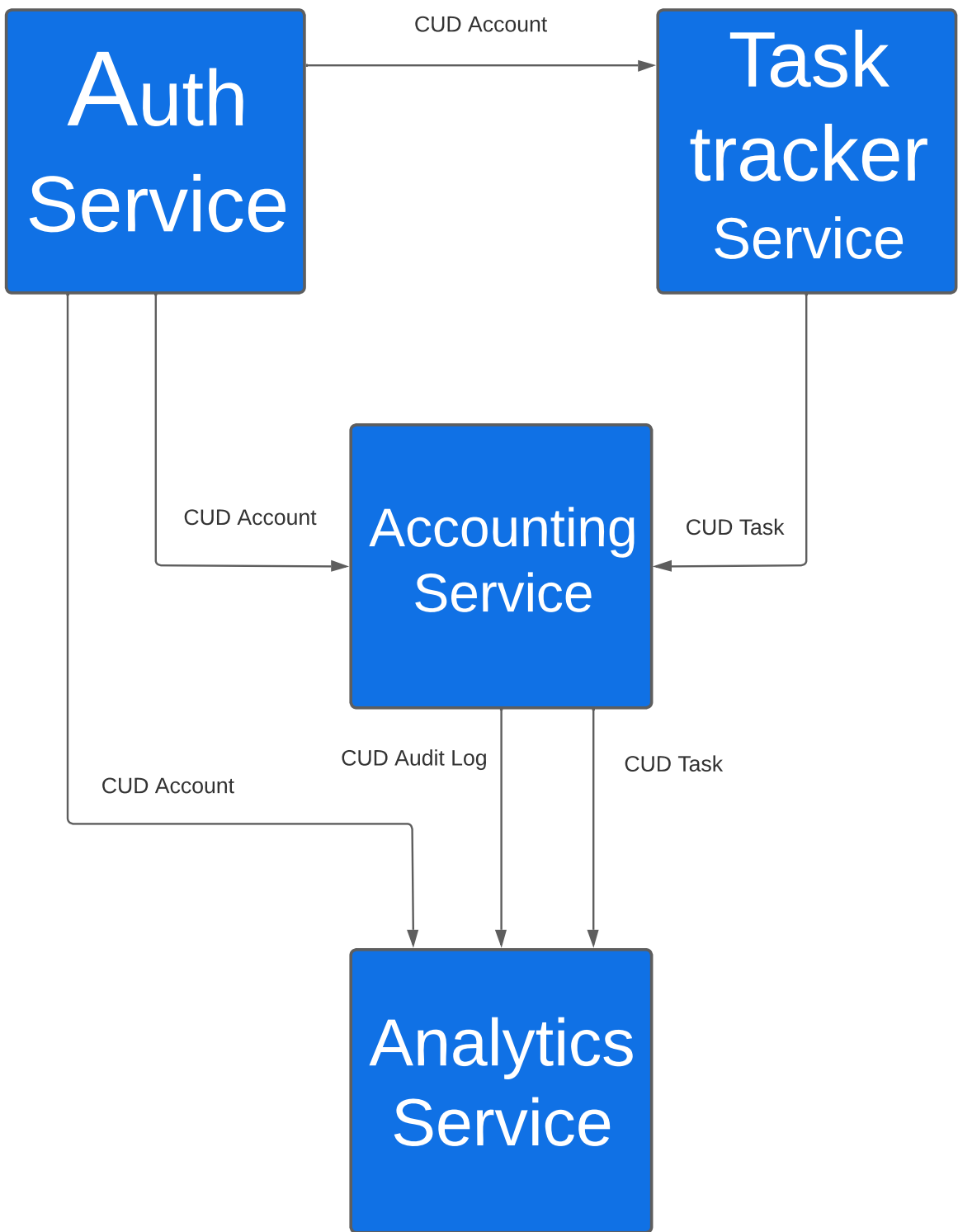
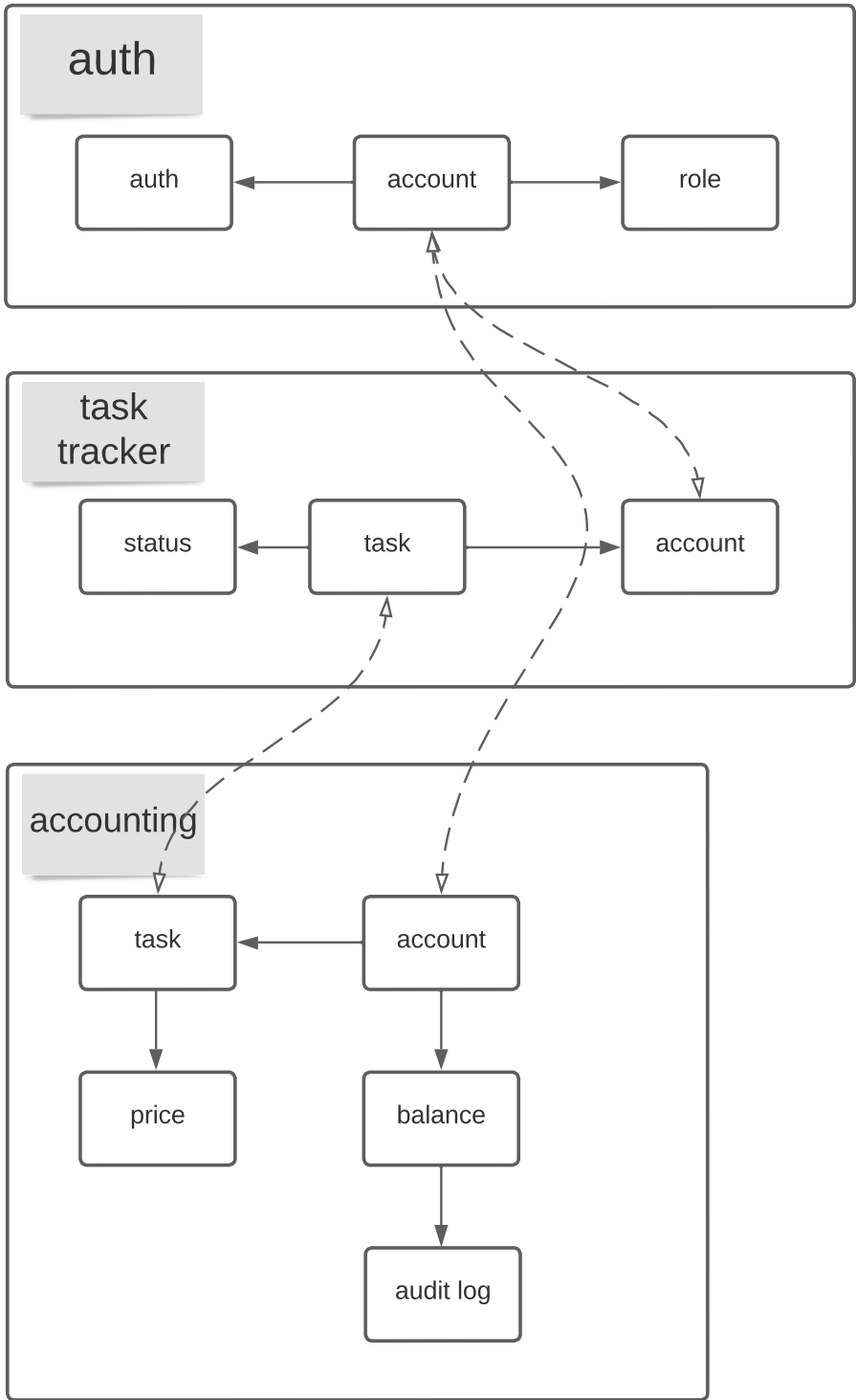
Actor - EndOfaDayCron
Command - Calculate today's profit
Data - Account public id + Au
Event - Accaunting.EndOfaDayExecuted

До разбора домашки во вторник Price у меня был отдельными данными в аккаунтинге, которые создавались на событие создания задачи, но теперь понятно, что смысла в этом нет и это не бизнес а cud, а цена часть Task в аккаунтинге. Точно то же самое с Аккаунтом и Балансом. С аудит-логом, ошибочно думал, что это бизнес событие, но это технический шаг до изменения баланса.

Analytics

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

Я подумал, что можно принимать единственный бизнес ивент о конце дня, для сохранения самой дорогой задачи и заработка менеджмента за сегодня (ну и очистка лишних данных, например). А в остальном просто собирать данные из CUD стриминга и по ним уже для пользователей любые запросы строить. При этом нам тут ен нужен баланс, по текущим условиям, но нужны события аудит лога и задачи уже с ценами.



- Actor
- Command
- Event
- Read Model

