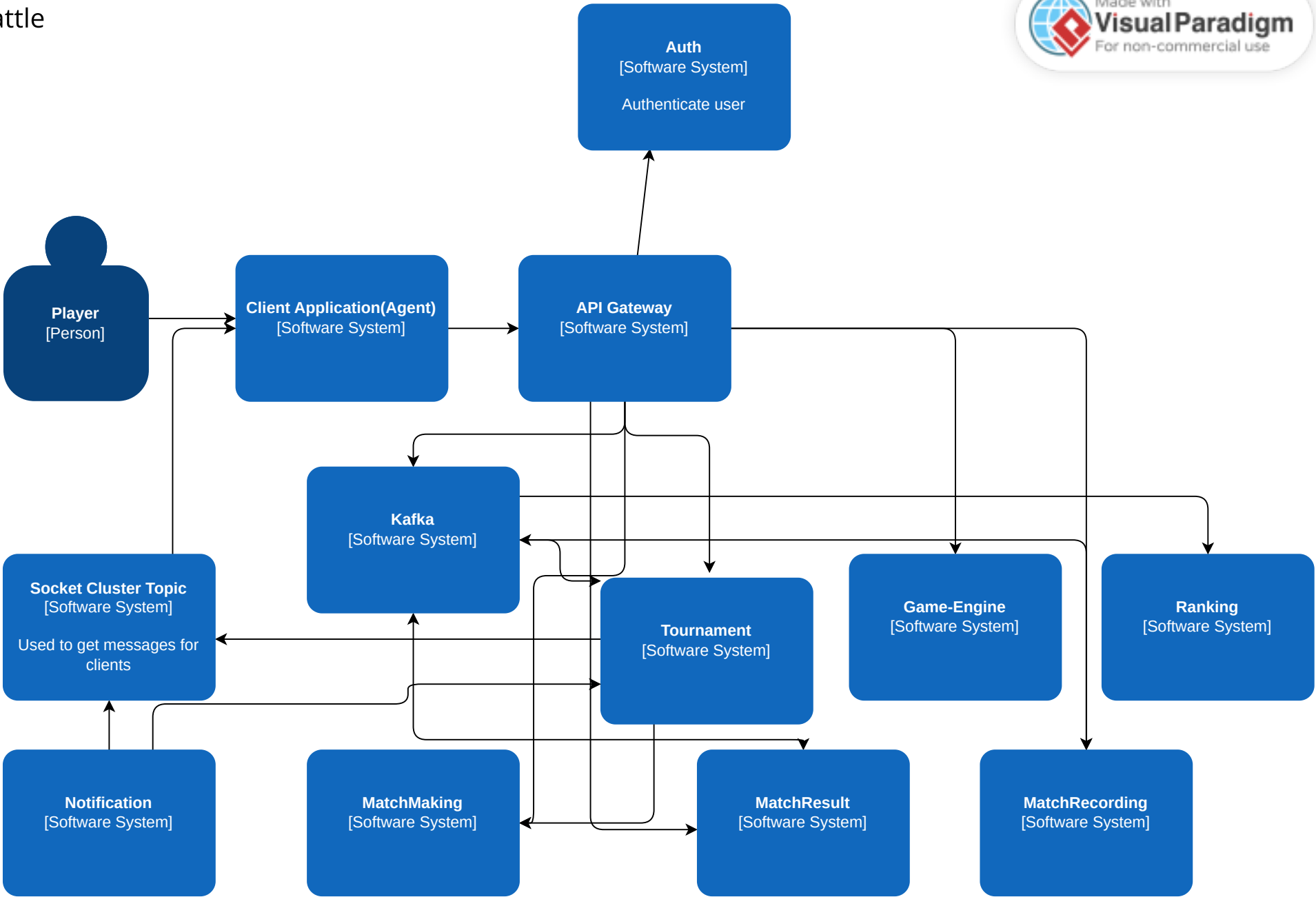


C3 diagram of Tank battle



Tournaments API:

- 1) POST /tournaments (tournament name, type, date, nullable(interval)) - SYNC
- 2) invite players(tournament name, playerIds) - ASYNC
- 3) approve players(tournament name, date, playerIds) - ASYNC
- 4) GET /tournaments (?startsAfter= etc.) SYNC
- 5) applyForParticipation(Tournament name, (optional) date) ASYNC
- 6) createMatch(TournamentId) ASYNC

Listen to RatingUpdatedEvent -> update rating for a tournament

Tournaments Events(Published to corresponding topics):

- 1) Tournament Created (tournament info)
- 2) PlayersInvited (players, tournament info)
- 3) ApprovedForTournamentEvent(player, tournament info)
- 4) DeclinedForTournamentEvent(player, tournament info)
- 4) ApprovedInvitationEvent(Player, tournament info)
- 5) DeclinedInvitationEvent(Player, tournament info)
- 6) AppliedForTournamentEvent(player, tournament info)

MatchMakingService

API

1. CreateMatch(Players, type) SYNC/ ASYNC



Ranking Service

Listens for:

- 1) ApprovedForTournamentTopic/ ApprovedInvitationTopic
-> increase tournament rank
- 2) LeaveTournamentTopic -> decrease tournament rank
- 3) MatchResultCalculated -> update player's ranks

Notification Service

Subscribes to:

- 1) PlayerInvitedTopic
- 2) ApprovedForTournamentTopic
- 3) DeclinedForTournamentTopic
- 4) AppliedForTournamentTopic
- 5) ApprovedInvitationTopic
- 6) PlayerDeclinedTopic
- 7) TournamentCreatedTopic
- 8) GameFinishedTopic

Notify players/services via messaging/email/application notificatoins

MatchResultService

Listens for:

- 1) Kill/Death/AssistEvent -> update matchResult
- 2) GameFinishedEvent -> send MatchResultCalculated
- 3) FinishedRecordedEvent -> update recordId

MatchRecording

Listens for:

- 1) GameStartedEvent -> startRecording
- 2) GameFinishedEvent -> stopRecording -> FinishedRecordedEvent
- 3) GET /videos/{id} - return presigned url

GameEngine

- 1) POST /operate/games/{id}/player/{id} - operate