ParkLogic
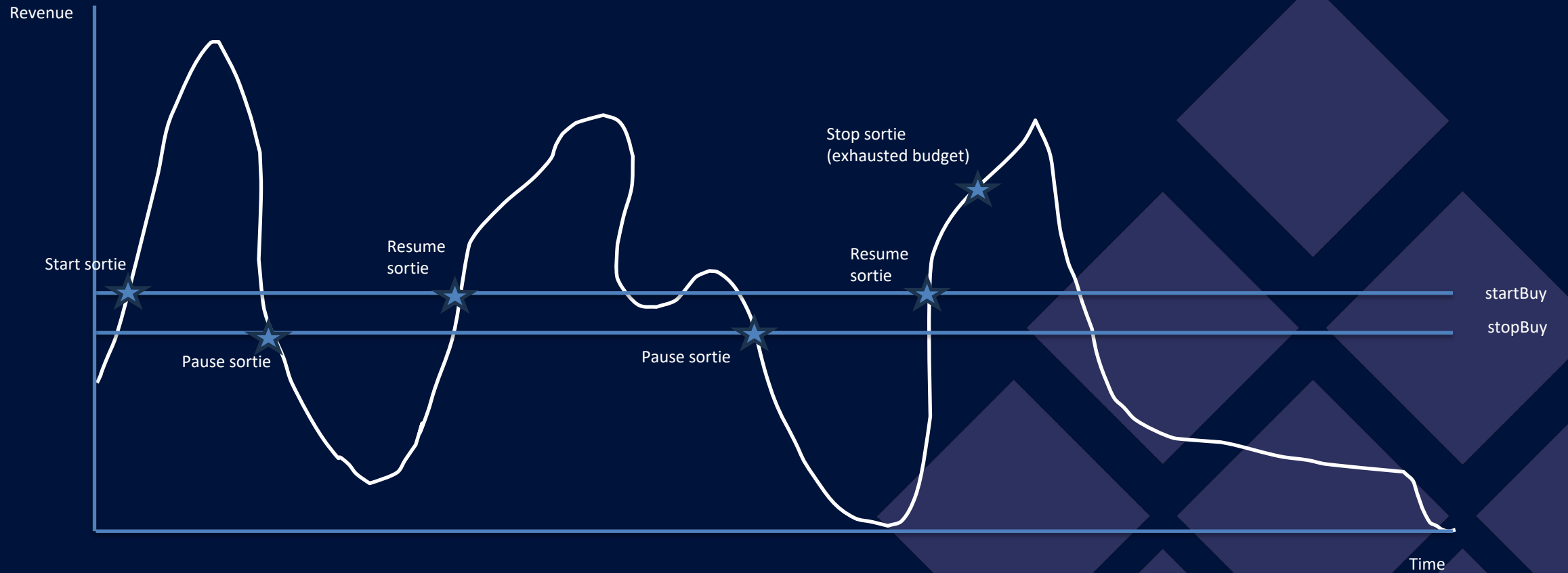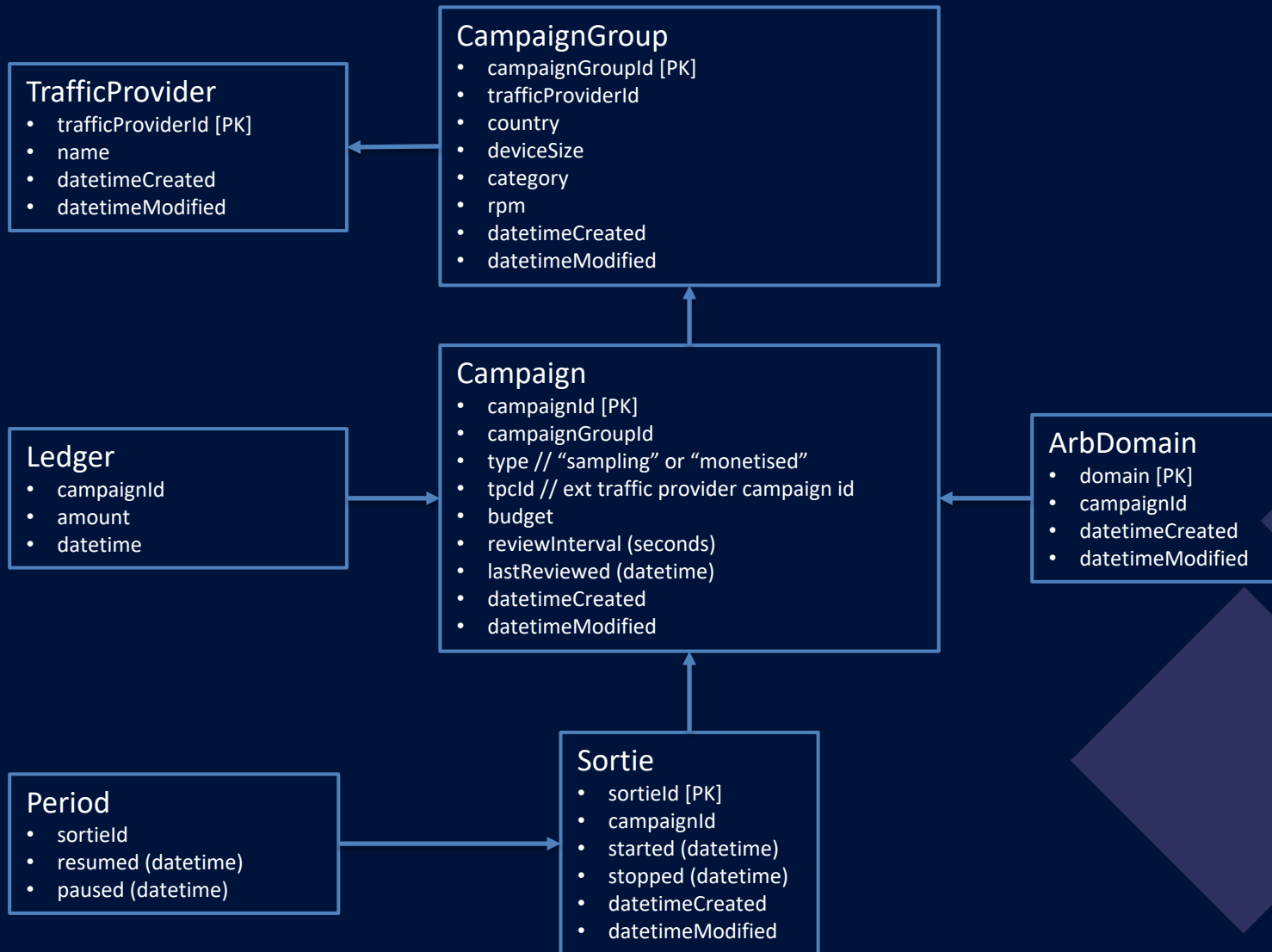
Changing the world together

Arbitrage System - Hackathon

Trend for a sampling campaign and responses

Database Tables

TrafficProvider
- trafficProviderId [PK]
- name
- datetimeCreated
- datetimeModified

CampaignGroup
- campaignGroupId [PK]
- trafficProviderId
- country
- deviceSize
- category
- rpm
- datetimeCreated
- datetimeModified

Ledger
- campaignId
- amount
- datetime

Campaign
- campaignId [PK]
- campaignGroupId
- type // "sampling" or "monetised"
- tpcId // ext traffic provider campaign id
- budget
- reviewInterval (seconds)
- lastReviewed (datetime)
- datetimeCreated
- datetimeModified

ArbDomain
- domain [PK]
- campaignId
- datetimeCreated
- datetimeModified

Period
- sortieId
- resumed (datetime)
- paused (datetime)

Sortie
- sortieId [PK]
- campaignId
- started (datetime)
- stopped (datetime)
- datetimeCreated
- datetimeModified

Campaign Notes
- Allocated Arb domains is based upon budget.
- We distribute traffic across multiple domains. The amount of revenue generated on an individual ArbDomain must be < $20.
- ArbDomain allocated to a Campaign remain with that Campaign.
- Campaign balance is retrieved via an API call
- Campaign url is set/retrieved via API
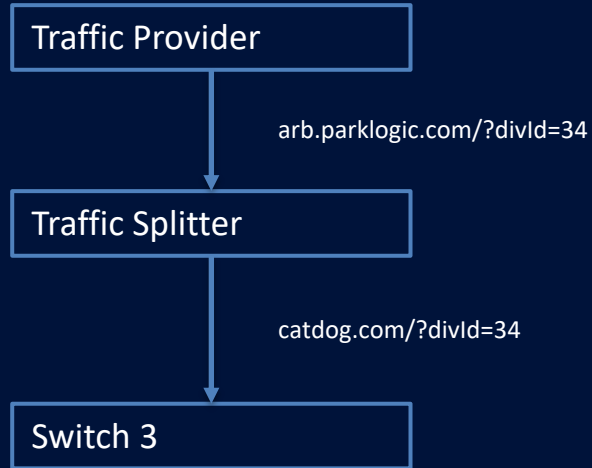
Pseudo Code - Processes

Sampling Campaign Review Process:

- Every minute: For all SampleCampaigns (SC)
    - If lastReviewed >= SC.reviewInterval seconds ago then
        - balance = get current balance from TPC
        - If balance < SC.budget then
            - Set balance to SC.budget
            - Record in Ledger table
        - SC. lastReviewed = NOW()

Monetised Campaign Review Process:

- Every minute:
- For all Monetised Campaigns (MC)
    - If no current sortie then
        - buyRPM = MC.rpm
        - sellRPM = select from CH DB average of last 5 winning bids where trafficProvider = PC.tpId and campaignId = MC.id
        - buyFactor = 1.2
        - If sellRPM > buyRPM * buyFactor then
            - If MC.balance < MC.budget then
                - Set MC.balance = MC.budget
            - Record in Ledger table
            - If successful set then
                - Start Sortie (max 1 per day)
    - Else // current sortie is running
        - depletedThreshold = 2 // US Dollar amount
        - If balance() < depletedThreshold then
            - Stop Sortie
        - Else
            - If sortiePaused then
                - If sellRPM > buyRPM * buyFactor then
                    - Resume sortie
            - Else // sortie running
                - sellFactor = 1.0
                - If sellRPM < buyRPM * sellFactor
            - Pause sortie

Traffic Flow

Traffic Provider

arb.parklogic.com/?divId=34

Traffic Splitter

catdog.com/?divId=34

Switch 3

Traffic Splitter Process:
- Select a random domain from arbDomains where divId=?
- Send 302 redirect to domain/?divId=34
- (This will send the traffic through to the switch3.)

Interface
- CRUD Traffic Provider
- CRUD Campaign Group
- Reports (TBC)

Traffic Provider APIs
- TBC

Questions
- How much to we add to the Traffic Provider balance.
- How much to we add to the Budget for each Campaign?
- Alerts for requiring additional funds for a Traffic Provider?

Arbitrage - Right to left algorithm

Campaigns point to a single point that has multiple domains behind it.

- *Campaign that sampes geo/device/UA for a specific price point and traffic provider*
- *Near real-time data from switch 3*
- *Provides RPM*

- *CPM = RPM / %margin*
- *Estimated revenue*
- *Budget*
- *< $20 per domain*

- System built – right to left
- Min profit $100K/month
- Apogee is the interface

Flowchart nodes:

Pool < 100 — Yes → Add new domains — No → Do nothing

Margin to be made

Stop campaign

Domain pool

Sampling campaign RPM — Provider Geo/Device/UA RPM → Activate campaign? — Yes → Keep Active? — No (Stop campaign) — Yes → Add/subtract domains to the campaign redirect → Turn on campaign at the provider → AdMaven, PopAds, Popcash, ExplorAds → Filter for bad traffic → Real-time data from switch 3

Real-time data from switch 3

Real-time data from switch 3