



History



Discover



Spaces



Finance



More



Notifications

Account

Install

Can you brainstorm me the idea of a software company which can optimize the ad spending on Facebook and google. Remember Facebook itself has ads optimization to increase the ROI. How would you do this ?

Reviewed 15 sources >

You can absolutely build a company here, but you should not try to "beat" Meta/Google's auction algorithms directly. Instead, you wrap them with better data, better objectives, and cross-channel decisioning that the platforms do not natively offer. [feedcast +1](#)

Below is a concrete product concept and how it would work.

Core product thesis

Position the company as a cross-channel performance layer that:

- Feeds the platforms higher-quality signals (conversion, value, LTV) than advertisers can today. [firstlaunch +1](#)
- Optimizes at the business level (profit, LTV, inventory, margin) instead of per-platform CPA/ROAS.
- Automates tedious ops (budget shifts, testing, negative signals) on top of Google/Meta's own automation. [zapier +1](#)

Think "smart CDP + bidding brain + experimentation engine" that orchestrates Facebook and Google, not replaces them. [segment +1](#)

Where Meta/Google are weak

You create value by attacking their blind spots, not their strengths:

- They only see on-platform behavior and last-click/attribution-window conversions unless you send more data. [firstlaunch +1](#)
- Smart bidding breaks when tracking is messy, goals are misconfigured, or conversion volume is low. [firstlaunch](#)
- They optimize against the signals you give them (e.g., button clicks vs revenue), not true business outcomes. [firstlaunch](#)
- They don't coordinate budgets across platforms around a unified objective (e.g., profit per day).

This leaves gaps: e.g., Meta might aggressively chase low-quality leads because "lead submit" is set as the main conversion, while actual SQLs or paid users are never fed back. [firstlaunch](#)

System design: 4 pillars

Imagine this as your V1 SaaS stack.

1) Data and attribution layer

- Collect first-party events from web/app (pixel/SDK, server-side tracking) with consistent IDs. [segment](#)
- Join with backend data: revenue, refunds, subscription renewals, product margins, LTV models.

- cost or goods, margin
- Use this to compute a normalized "true value per impression" signal per platform and campaign.

You are effectively Segment/Twilio Segment style activation but opinionated for ads optimization. [segment](#)

2) Smart signal generation back to platforms

Instead of letting Meta/Google optimize to weak proxies, your product translates real outcomes into clean, high-signal events: [segment +1](#)

- Define a small set of platform-facing events:
 - "High value purchase" (e.g., predicted LTV > threshold)
 - "Qualified lead" (meets CRM qualification rules)
 - "Churn risk prevented" (for reactivation campaigns)
- Map your internal events → these conversion types, with appropriate values.
- Push them server-to-server (CAPI for Meta, enhanced conversions + offline conversions for Google) with real or modeled conversion values. [segment +1](#)

Your differentiation: easy setup, guardrails, best-practice default mappings per vertical.

3) Cross-channel budget and bidding brain

You don't set bids inside auctions, but you do decide:

- How much daily/weekly budget each platform, campaign, and audience should get.
- Which bidding strategy target (tCPA/tROAS) to use given performance and volume. [learn.jyli +1](#)

Approach:

- Ingest performance from Google Ads, Meta, and analytics (clicks, spend, conversions, revenue, LTV). Tools like Feedcast/AdScale and Revealbot already show the appetite for this type of centralization and automation. [mt-project +1](#)
- Compute marginal return curves: "If I add \$X to Campaign A on Meta vs Campaign B on Google, which yields more profit/ROAS?"
- Implement a daily (or intra-day) optimizer that:
 - Shifts budget from underperforming to overperforming campaigns across platforms.
 - Adjusts tROAS/tCPA targets within safe bounds.
 - Pauses segments with negative marginal return.

You can start with simple rules, then evolve to bandit-style or RL-based policies as you get more customers and data.

4) Automation and experimentation engine

On top of that brain, ship automations that marketers already know they should do but rarely have time for. Existing tools like Revealbot and AdScale validate demand here. [feedcast +1](#)

Ask a follow-up



Data structures

```
python
# One row per adset / campaign on each platform
Ask a follow-up
@dataclass
revenue      # attributed revenue in same window
conversions
clicks
impressions

# Derived metrics
@property
def roas(self):
    return self.revenue / self.spend if self.spend > 0 else 0.0

@property
def cpa(self):
    return self.spend / self.conversions if self.conversions > 0 else
float("inf")
```

1) Read data from platforms

```
python
def fetch_arm_states(time_window):
    fb_rows = fb_api.get_insights(time_window)      # per adset [web:19][web:26]
    goo_rows = google_api.get_insights(time_window) # per campaign [web:24]
    [web:30]

    arms = []
    for r in fb_rows:
        arms.append(ArmState(platform="facebook", id=r.adset_id, ...))
    for r in goo_rows:
        arms.append(ArmState(platform="google", id=r.campaign_id, ...))
    return arms
```

2) Simple bandit-style allocator

You estimate a score per "arm" (adset/campaign) and reallocate budget.

```
python
def score_arm(arm: ArmState, min_conversions=5):
    # If very low data, use exploration bonus
    if arm.conversions < min_conversions:
        return 1.0 # small positive score to keep exploring
    return arm.roas # or expected_profit(arm) if you model LTV

python
def allocate_budget(arms, total_budget):
    # Compute non-negative scores
    scores = {a.id: max(score_arm(a), 0.0) for a in arms}
    score_sum = sum(scores.values())
    if score_sum == 0:
        # Fallback: equal split
        share = scores[a.id] / score_sum
        budgets[a.id] = total_budget * share
        return budgets

    Ask a follow-up
@dataclass
revenue      # attributed revenue in same window
conversions
clicks
impressions

# Get current budget from platform
current = get_current_budget(arm)      # via FB/Google APIs [web:20][web:24]
[web:33]

# Clamp change to avoid wild swings
upper = current * (1 + max_change_ratio)
lower = current * (1 - max_change_ratio)
target = min(max(new_daily, lower), upper)

if abs(target - current) / current < 0.05:
    continue # ignore tiny changes

if arm.platform == "facebook":
    fb_api.update_adset_budget(adset_id=arm.id, daily_budget=target) #
[web:19][web:26][web:32]
else:
    google_api.update_campaign_budget(campaign_id=arm.id,
daily_budget=target) # [web:24][web:30][web:33]
```

3) Turn allocation into API calls

```
python
def apply_budget_changes(arms, new_budgets, max_change_ratio=0.3):
    for arm in arms:
        new_daily = new_budgets[arm.id]

        # Get current budget from platform
        current = get_current_budget(arm)      # via FB/Google APIs [web:20][web:24]
        [web:33]

        # Clamp change to avoid wild swings
        upper = current * (1 + max_change_ratio)
        lower = current * (1 - max_change_ratio)
        target = min(max(new_daily, lower), upper)

        if abs(target - current) / current < 0.05:
            continue # ignore tiny changes

        if arm.platform == "facebook":
            fb_api.update_adset_budget(adset_id=arm.id, daily_budget=target) #
[web:19][web:26][web:32]
        else:
            google_api.update_campaign_budget(campaign_id=arm.id,
daily_budget=target) # [web:24][web:30][web:33]
```

4) Full daily optimization loop

```
python
def optimize_once(account_id, total_budget):
    arms = fetch_arm_states(time_window="yesterday")
    new_budgets = allocate_budget(arms, total_budget)
    apply_budget_changes(arms, new_budgets)

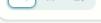
    # Cron: run every night (or every few hours)
    while True:
```

```
optimize_once(account_id="123", total_budget=100000) # 100k/day example
sleep(24 * 60 * 60)
```

5) Feeding better conversion signals

Separately, you send high-quality conversion back to both platforms.

Ask a follow-up



```
        "value": value,
        "event_name": "high_value_purchase" if is_high_ltv else "purchase"
    }

    fb_capi.send_conversion(event)           # Meta CAPI [web:8]
    google_offline.send_conversion(event)   # Google offline / enhanced
    conversions [web:7][web:24]
```

Your "software company" is basically:

- The scheduler around `optimize_once`.
- The data plumbing that builds `ArmState` from many sources.
- The smarter `score_arm` (profit, LTV, bandits/RL) instead of my simple ROAS.

If you want, I can rewrite this as a more RL-style loop (state, action, reward) with clearer exploration logic.

15 sources

How would you prove that the idea works

Reviewed 15 sources >

You prove it works the same way serious advertisers prove any optimization works: with controlled experiments that show incremental lift, not just nicer dashboards. [facebook +1](#)

1. Define a clear causal question

- Example: "Does our optimizer increase profit / ROAS vs letting Meta + Google run on their own, holding total spend constant?"
- Pick 1-2 primary metrics: incremental revenue, incremental profit, or incremental ROAS ($\text{incremental revenue} \div \text{spend}$). [thinkwithgoogle +1](#)

2. Design A/B or geo experiments

Account-level / campaign-level split

- Take a set of similar campaigns/adsets.
- Randomly assign half to "control" (pure platform automation) and half to "treatment" (your optimizer controlling budgets/bids).
- Keep targeting, creatives, and total spend as similar as possible; only the decision policy differs. [heus +1](#)

Geo holdout (GeoLift-style)

- Split regions into test vs control geos using tool like GeoLift or similar methodology.

Ask a follow-up



3. Run long enough for statistical power

- Use historical performance to estimate variance and minimum detectable effect, then compute needed duration (often 2-6 weeks per test). [incremental +1](#)
- Lock in the test plan: no mid-test strategy changes, budgets swings, or creative overhauls that confound results. [facebook +1](#)

4. Compute incremental lift

For each test:

- Measure outcomes in control vs treatment:
 - Control revenue: R_c , Treatment revenue: R_t , Control spend: S_c , Treatment spend: S_t .
- Compute:
 - Incremental revenue $\approx (R_t - R_c)$ after normalizing for baseline differences.
 - Incremental ROAS $\approx (R_t - R_c)/S_t$. [measured +1](#)
- Use standard stats (t-test or Bayesian credible intervals) to show significance of the lift. [incremental +1](#)

5. Layer bandit / RL validation

- Within treatment, you can also run multi-armed bandit experiments (e.g., UCB/Thompson sampling vs your heuristic allocator) to show your policy beats simpler policies while learning faster. [towardsdatascience +1](#)
- Again: hold total spend constant and compare incremental profit/ROAS between policies.

6. Productizing the proof

Your "proof engine" can be a feature of the product:

- Self-serve test setup wizards (choose metric, pick campaigns/geos, auto-assign control vs treatment). [workmagic +1](#)
- Built-in incrementality calculator and report: "Your optimizer drove +18% incremental ROAS and +12% incremental profit at 95% confidence over 28 days." [thinkwithgoogle +1](#)

This way, every new client gets a scientifically clean, time-bound test that either validates the optimizer or tells you where it fails.

15 sources

how would you get data from facebook and google platforms. Show some real life api example

Reviewed 15 sources ↴

Ask a follow-up

Search icon, Refresh icon, Print icon, Copy icon, Up arrow icon, Down arrow icon, Home icon, Help icon

1. Facebook / Meta Ads – get Insights via Python SDK

Typical flow: init SDK with access token, then query the Insights endpoint at account or campaign level. [developers.facebook +1](#)

```
python
from facebook_business.api import FacebookAdsApi
from facebook_business.adobjects.adaccount import AdAccount
from facebook_business.adobjects.adsinsights import AdsInsights

APP_ID = "YOUR_APP_ID"
APP_SECRET = "YOUR_APP_SECRET"
ACCESS_TOKEN = "YOUR_LONG_LIVED_ACCESS_TOKEN"
AD_ACCOUNT_ID = "act_1234567890"

FacebookAdsApi.init(APP_ID, APP_SECRET, ACCESS_TOKEN)

account = AdAccount(AD_ACCOUNT_ID)

fields = [
    AdsInsights.Field.campaign_id,
    AdsInsights.Field.campaign_name,
    AdsInsights.Field.impressions,
    AdsInsights.Field.clicks,
    AdsInsights.Field.spend,
    AdsInsights.Field.actions,
]

params = {
    "date_preset": "yesterday",      # or "last_7d", or custom time_range
    "level": "campaign",           # can be adset / ad / account
    "time_increment": 1,            # daily breakdown
}

insights = account.get_insights(fields=fields, params=params)

for row in insights:
    print(
        row[AdsInsights.Field.campaign_id],
        row[AdsInsights.Field.campaign_name],
        row[AdsInsights.Field.impressions],
        row[AdsInsights.Field.clicks],
        row[AdsInsights.Field.spend],
    )
}
```

- This is basically what many guides and StackOverflow examples do to grab campaign stats. [stackoverflow +2](#)

Ask a follow-up

Search icon, Refresh icon, Print icon, Copy icon, Up arrow icon, Down arrow icon, Home icon, Help icon

You can either update budgets directly on ad sets, or define automated rules. Meta's docs show an example `CHANGE_BUDGET` rule via curl. [developers.facebook +1](#)

```
bash
curl \
  -F 'name=Decrease budget for high-frequency ad sets' \
  -F 'schedule_spec={
    "schedule_type": "CUSTOM",
    "schedule": [{"start_minute":0,"days":[2,5]}]
  }' \
  -F 'evaluation_spec={
    "evaluation_type": "trigger",
    "filters": [
      {"field": "impressions", "value": 8000, "operator": "greater_than"},
      {"field": "frequency", "value": 5.0, "operator": "greater_than"}
    ]
  }' \
  -F 'execution_spec={
    "execution_type": "CHANGE_BUDGET",
    "execution_options": [
      {"field": "change_spec",
       "value": {"amount": -30, "unit": "PERCENTAGE"}, 
       "operator": "EQUAL"}
    ]
  }' \
  -F "access_token=<ACCESS_TOKEN>" \
  https://graph.facebook.com/v19.0/<AD_ACCOUNT_ID>/adrules_library
```

For your own optimizer, you'd usually call the ad set endpoint from Python instead of rules, but this shows the real parameters Meta expects for budget changes. [developers.facebook +1](#)

3. Google Ads API – query stats with GAQL in Python

Google Ads uses GAQL (Google Ads Query Language) to fetch metrics. [github +1](#)

```
python
from google.ads.googleads.client import GoogleAdsClient

# Reads google-ads.yaml from disk (developer token, OAuth, etc.)
client = GoogleAdsClient.load_from_storage(version="v20")
```

```

customer_id = "1234567890" # without dashes

ga_service = client.get_service("GoogleAdsService")

query = """
    SELECT
        campaign.id,
        campaign.name,
        metrics_impressions
"""

Ask a follow-up

```

- This mirrors Google's official sample for `get_campaigns.py`, just with extra metrics.
- Again, you'd normalize this into your `ArmState` objects.

4. Google Ads – update campaign daily budget

To let your optimizer change budgets, you call the `CampaignBudgetService` (or shared budgets) with an update operation.

```

python
from google.ads.googleads.client import GoogleAdsClient
from google.ads.googleads.v20.resources.types import CampaignBudget
from google.ads.googleads.v20.services.types import CampaignBudgetOperation
from google.protobuf import field_mask_pb2

client = GoogleAdsClient.load_from_storage(version="v20")

customer_id = "1234567890"
budget_service = client.get_service("CampaignBudgetService")

# Existing budget resource name, e.g. "customers/1234567890/campaignBudgets/111"
budget_resource_name = budget_service.campaign_budget_path(customer_id, "111")

new_amount_micros = int(500 * 1_000_000) # 500.00 in account currency

budget = CampaignBudget(
    resource_name=budget_resource_name,
    amount_micros=new_amount_micros,
)

```

```

Ask a follow-up

```

```

response = budget_service.mutate_campaign_budgets(
    customer_id=customer_id,
    operations=[operation],
)

print("Updated budget:", response.results[0].resource_name)

```

This is the kind of call your `apply_budget_changes` step would make once your bandit/RL logic has decided new target budgets.

If you want, next we can plug these snippets into the earlier pseudo code so you see a single end-to-end script: fetch from Meta + Google, normalize to a table, compute scores, then push back new budgets.

16 sources

are there any new startups working on this idea ?

Reviewed 15 sources >

Yes—there are already startups and tools circling this idea, which is both validation and a hint about how you should differentiate.

Existing tools in this space

- Revealbot – automates rules, budget changes, and bid tweaks for Meta and other platforms, positioned as an AI assistant to increase ROI on ads.
- BudgetFlow AI (example from a 2025 tools roundup) – “spend & bidding automation” that reallocates budget across campaigns and channels in real time using ML, very close to the bandit/optimizer idea.
- Cross-channel optimization platforms like Keen – focus on allocating budgets across multiple paid channels using ML and incremental data, not just reporting.

- AI ad platforms like Albert.ai – used by brands to automatically manage and optimize cross-channel ad spend (e.g., reported 40% reduction in cross-channel spend for some clients while improving results). [xeriangrowthagency](#)

What this means for your idea

- The category is real and funded: cross-channel, AI-driven budget and bidding optimization is an acknowledged need; tools already promise real-time reallocation, predictive bidding, and ROI uplift. [keends](#) +2
- The gap is in how you do it and for whom: [down](#), deeper LTV-based optimization for a niche

Ask a follow-up



[15 sources](#)

[15 sources](#)

[15 sources](#)

how can I get funding on this idea ?

Reviewed 15 sources >

Investors will fund this kind of product, but only after you show sharp positioning and early proof that agencies/brands want it. [growthlist](#) +1

1. Tighten the story for investors

- Frame it as AI-native, vertical B2B SaaS: "LTV- and profit-based budget optimizer for Meta + Google for e-commerce/SaaS advertisers." AI + vertical SaaS is exactly what's getting funded. [saastock](#) +1
- Clarify the wedge: e.g., "We plug into ad accounts in 1 hour and recover 10–20% wasted spend by fixing goals + cross-channel allocation, proven via built-in incrementality tests."

2. Validate before real fundraising

- Talk to 20–30 agencies / performance marketers and ask for pain, workflow, and willingness to pay; B2B SaaS guides emphasize interviews and early design partners as key signals. [nomadicsoftware](#) +2
- Build a scrappy MVP: a script + dashboard that runs on 1–3 pilot accounts; charge even a small monthly fee to show real revenue, which seed investors look for. [foruvic](#) +1

3. Pick funding path by stage

- Pre-seed / angels: Pitch on problem, founder-market fit, and 2–3 paying pilots; typical SaaS pre-seed checks are in the low hundreds of thousands to around \$1–2M. [ecapabs](#) +1
- Seed: Aim once you have an MVP, reference customers, and early ARR; 2025 medians for SaaS seed were about \$2M with clear problem-solution fit. [growthlist](#) +1
- Accelerators: YC, Techstars, and martech-focused accelerators (e.g., Hawke Ventures-linked programs) are active in AI/martech and can be an easier first cheque plus customers. [elity](#) +1

4. Who to actually pitch

- Generalist SaaS / AI VCs: Menlo, Insight, a16z, etc., which have strong SaaS/AI portfolios, but harder to access early. [rba](#) +2
- Martech / commerce-tech specialists and corporate VCs: funds like Hawke Ventures, Martech Ventures, Salesforce Ventures, Adobe Ventures focus on marketing and analytics tools and bring distribution. [vc-mapping-gillion](#) +1
- Local angle (you're in Seattle): Madrona is [down](#) explicitly focused on Pacific Northwest

Ask a follow-up



- Product + demo: screenshots or short Loom of your optimizer reallocating budget and the lift it produced.
- Traction: pilot customers, \$X/month in ARR, case study like "Agency Y got +15% ROAS in 6 weeks at constant spend."
- Go-to-market: start with agencies that manage many Meta/Google accounts; this is aligned with B2B SaaS VC expectations for early GTM clarity. [foruvic](#)

If you share where you are today (no code, prototype, pilots), I can outline an exact 3–6 month "funding prep" plan tailored to you.

[15 sources](#)

[15 sources](#)

[15 sources](#)

I am currently on H1B visa. After I get the funding for pre-seed, can I apply for O1 visa

Reviewed 15 sources >

Yes, you can apply for an O-1 after raising a pre-seed, but funding alone is not enough; you still have to meet the O-1 "extraordinary ability" criteria in business/technology. [boundless](#) +1

1. Role of funding in an O-1 case

- Seed/pre-seed from reputable VCs or angels is positive evidence that your startup and your role are "distinguished" and important; lawyers routinely use funding as part of the proof package. [calivisa](#) +2
- USCIS does not treat venture funding by itself as a qualifying "award"; recent guidance says funding should mainly support criteria like critical role, original contributions, or company reputation. [opensphere](#) +2

2. What you actually need to qualify

For O-1A (business/tech), you must meet at least 3 of 8 regulatory criteria, such as: [h1b.harvard](#) +1

- Major awards or significant prizes in your field.
- Memberships that require outstanding achievements.
- Press / media about you or your work (articles, interviews, tech press, etc.).
- Original contributions of major significance (e.g., widely used product, patents, recognition letters from experts). [manifestlaw](#) ↴
- Critical or essential role for a distinguished organization (here your funded startup + investors, customers, accelerators). [greenacardlink](#) ↴
- High remuneration (salary, equity value) compared to peers.

Funding can help prove: company reputation, critical role, and sometimes

Ask a follow-up

* Form a U.S. entity and make it the founder (often via a board-independent director setup to show a real employer-employee relationship). [rlawgroup](#) ↴

- Raise pre-seed from recognized investors or join a reputable accelerator; use that plus press and letters from well-known people to show your startup is distinguished and your role is critical. [deel](#) ↴
- Collect evidence now: talks, open-source, patents, awards, media mentions, judging hackathons, etc.—founder-focused guides emphasize building this “evidence portfolio” early. [relegate](#) ↴

4. Key caveats

- Being on H-1B does not block you from applying for O-1, but you **must** keep lawful status during the transition (timing with your lawyer is important).
- Acceptance into an accelerator or getting VC backing **does not automatically qualify** you for O-1; it's strong supporting evidence, not a guarantee. [beyondborderglobal](#) ↴

Because standards have tightened, especially for AI/tech founders, it's smart to speak with a U.S. immigration attorney who specializes in O-1 for startup founders and show them your CV + planned funding; they can tell you how close you are and what evidence to build over the next 6–12 months. [jspvise](#) ↴

15 sources

can you provide good names for this company

Here are some name ideas oriented around AI, optimization, and ad spend control:

Performance / ROI focused

- ProfitPilot
- SpendSense
- ROASFoundry
- MarginMind
- ProfitCircuit
- YieldEngine

Optimization / brain / AI angle

- OptiBrain
- BidNeuron
- VectorSpend
- GradientLabs
- BanditIQ
- PolicyFlow

Cross-channel / orchestration vibe

- OrbitSpend
- ChannelCraft
- CrossBeam AI
- OmniBudget
- NexusOptim
- MultiLift

More brandable / abstract

- Advera Labs
- Zentrova
- Luminance AI
- Fluxmetric
- Elevio Ads

If you tell me your preferred style (serious B2B vs playful, short vs descriptive, .com availability), I can narrow this to 5–6 strong contenders with quick taglines.

Ask a follow-up