Assumptions

- Monthly data; impact lands after a learned lag (usually 2–3 months).
- Optimize for pipeline uplift; caps enforced in the partner split.
- Budget B is fixed; a small slice is reserved for exploration.

What each agent does (one-liners)

Agent	**Input → Output**	**What it does**
Lag Learner	Grants, Outcomes → L	Finds the delay (L) that best links “spend at t” to “pipeline at t+L” by correlation.
ROI Learner (campaign)	Grants aligned by L, Outcomes → posterior(ROI_campaign)	Builds a safe ROI estimate per campaign type (Gamma posterior from historical pipeline and spend).
Campaign Allocator	posterior, B → per-campaign allocation + uplift	Thompson samples ROI to get prob. of being best, allocates budget (with exploration), computes uplift = allocation × ROI.
Partner Fairness	Grants, Outcomes, Caps, B, L → top-K partners + fair split	Ranks partners by historical ROI (pipeline/spend, aligned by L), picks top-K, splits B fairly across them under caps, computes partner uplift.
Forecaster	Baseline, L, uplifts → forecast	Adds total uplift to the baseline in month m+L to show the impact.
Reviewer (LLM)	JSON summary of the above → plain-English report	Explains the decision, lag, totals, top campaigns, and the partner fairness logic in short paragraphs.

Step-by-step

1. **Load data**
    - Grants (month, partner, campaign, amount)
    - Outcomes (month, partner, pipeline, cap/tier)
    - Baseline next 3 months (month, partner, baseline_pipeline)
2. **Learn lag L**
    - Try L in {2,3}. For each, align grants at month t to outcomes at t+L and compute corr(spend, pipeline).
    - Pick the L with the highest correlation.
3. **Estimate campaign ROIs (posterior)**
    - Align data by lag: spend at t → pipeline at t+L.

    - Aggregate by **campaign type**: total_pipeline and total_amount.
    - Compute a **Gamma posterior** per campaign:
      post_alpha = prior_alpha + total_pipeline, post_beta = prior_beta + total_amount.
    - This gives a distribution of ROI_campaign (pipeline per $) that is stable on small data.

4. **Allocate by campaign (explore/exploit)**
    - **Thompson sampling**: draw ROI samples from each campaign posterior.
    - Probability of being best = share of draws where that campaign has the highest ROI.
    - Allocate **(1–ε)×B** proportionally to those probabilities; allocate **ε×B** to under-observed to learn.
    - **Campaign uplift** = allocation × sampled ROI.
    - Table 1 (natural headers):
        - "Campaign type", "Probability", "MDF allocation", "Pipeline uplift".

5. **Partner fairness layer (top-K)**
    - Compute **partner ROI** = (pipeline/spend), using the same lag alignment.
    - Rank partners by ROI; pick **top-K** (K=2 by default).
    - Split **B fairly** across those K partners (equal shares), respecting **cap_monthly**; redistribute leftovers to others under cap.
    - **Partner uplift** = allocation × partner ROI.
    - Table 2 (natural headers):
        - "Partner", "Baseline pipeline" (at m+L), "MDF allocation", "Pipeline uplift".

6. **Forecast impact**
    - Sum the selected uplift(s).
    - Identify **impact month** = current month + L.
    - **Forecast (with MDF)** = **Baseline** + **Total uplift** in that impact month.
    - Plot: Baseline vs With-MDF lines over the next 3 months.

7. **LLM report (gemma-3-27b-it)**
    - We pass a compact JSON: {lag, budget, impact month, total uplift, top campaigns (alloc+uplift), partner fairness split (top-K partners with alloc+uplift)}.
    - The LLM writes 2–3 short paragraphs:

        - What we learned (L), budget, impact month, total uplift.
        - Why these **campaigns** got budget (probability, exploration).
        - Why we **split across top-K partners** (high ROI + fairness + caps) and the expected uplift.

8. **Deliverables**
    - **PDF**: narrative + Table 1 (campaigns) + Table 2 (partners) + impact plot.
    - **CSV**: campaign allocation details.
    - Console: lag, impact month, total uplift.

Why the conclusions are credible

- **Lag-aware**: we attribute money at t to results at t+L.
- **Bayesian**: posteriors prevent overreacting to noisy small samples.
- **Explore/exploit**: we learn while we earn.
- **Fairness**: we avoid over-concentrating by splitting across the top performers, within policy caps.
- **Explainability**: natural-language report ties numbers to decisions.