RFC - <Service/Feature/Process Name>

mm/yy

## Instructions

1. Keep it concise! Not more than 10 pages before the Appendix.
2. READ THIS FIRST: [go/rfc](http://go/rfc) and [go/rfctips](http://go/rfctips) OR <Link to your team’s RFC wiki>
3. Define the problem you are trying to solve and send an Intent to RFC email to <team>-rfcs@linkedin.com using this [template](https://docs.google.com/a/linkedin.com/document/d/1kAS-3HHab5TQQJocFPECFT0W1Fm8NFIP_PrsIdoZjBU/edit?usp=sharing).
4. Select “File -> Make a copy…” and create a copy of this document titled “<Service/Feature/Process Name> RFC” and put in the <Team> RFCs [folder](https://drive.google.com/open?id=0B46Z25fFtKfBVW1KVUJOckN3UzQ).
5. If a particular section is not applicable to your feature simply write N/A or Not Applicable and explain why if it is not obvious. The template is simply to guide you, feel free to structure your RFC the way you would like it.
6. Delete this “Instructions” section.
7. Post a message announcing the new RFC to the RFC Reviewers Group: send an email to <team>-rfcs@linkedin.com. Subject: <Service/Feature/Process Name> RFC
   1. Before you post:
      1. Make sure you’ve joined the RFC Reviewers Group if not already on it
      2. Make sure your RFC is in the shared RFC folder on Google Docs
   2. In your post:
      1. Provide a link to your document
      2. Name your core reviewers, key stakeholders/PoC from other teams
8. As comments are added, please update the body of the RFC and resolve the comment when a solution is found. Collaboration is the key to success here. Encourage your reviewers to be [Open, Honest and Constructive](http://go/values).
9. Request your reviewer’s to write “LGTM” against their names once their feedback has been addressed. You can open an action item for them by highlighting their name in the reviewers section.

## Authors

* Bob (bob@linkedin.com)
* Alice (alice@linkedin.com)
* You are the R (and most likely one of the Ps) in the [RAPID framework](http://go/rapid).

## Eng Manager(s)/Tech Lead(s)/PM(s)

* Joe (joe@linkedin.com)
* Katy (katy@linkedin.com)
* Your tech lead or Eng Manager is the D in RAPID framework in most cases.

## Reviewers

| Name | Review Status |
| --- | --- |
| Jane (janeTeamLead@linkedin.com) |  |
| Frank (frankClientReviewer@linkedin.com) |  |
| Mary (marySamzaReviewer@linkedin.com) |  |

These are the A in the [RAPID framework](http://go/rapid). Any other reviewers are the I in the [RAPID framework](http://go/rapid).

## Key Stakeholder(s)

* Bob(bobEspressoPoC@linkedin.com)
* Alice([aliceSecurityPoC@linkedin.com](mailto:aliceSecurityPoC@linkedin.com))
* Infra Team 1
* SRE Team 2
* These are I (and also one of the Ps) in the [RAPID framework](http://go/rapid).

## Link to Implementation

Provide a link to the project homepage or a jira dashboard/epic that can be used to track the overall status of the project referenced in this RFC.

# Abstract

*Keep it concise! Not more than 10 pages before the Appendix.*

This section should provide a concise overview of this document. After reading this abstract, the reader should have a high-level understanding of why the problem is important (the "why"), the problem being solved (the "what"), and how this design intends to solve it (the "how"). When the draft design document is circulated, readers may use this abstract to determine if it is applicable to their area of work and if they are interested in reading further.

# Motivation

*Keep it concise! Not more than 10 pages before the Appendix.*

Please touch on “**why**” the problem is important or the motivation behind solving this problem just like you did in your [Intent to RFC](http://go/intentToRfc). Please provide some background in the form of a brief high-level summary of the current state and links to any eng/product/design specs that are needed to understand the problem and the solution.

# Problem Statement

*Keep it concise! Not more than 10 pages before the Appendix.*

This section should frame the problem clearly and state the non-negotiable requirements for a solution just like you did in your [Intent to RFC](http://go/intentToRfc). “**What**” should the solution solve for? There should be no disagreement on the problem statement, otherwise, a design or implementation is unlikely to be successful.

# Architecture/Implementation

*Keep it concise! Not more than 10 pages before the Appendix.*

This section must describe the proposed design and implementation in detail, and “**how**” it addresses the problems outlined in the problem statement. It must also provide details on alternative solutions that were considered, stating clearly why they lost out to the final design. Include architecture diagrams and flows in this section.

# Multi-colo concerns

*Keep it concise! Not more than 10 pages before the Appendix.*

What are the multi-colo implications for your service? Are traffic-shifts and failovers handled gracefully?

# Dependencies

*Keep it concise! Not more than 10 pages before the Appendix.*

Which services do you depend on and which services depend on you? Are there dependent projects/features? Fill in [DependIn](http://go/DependIn) ticket if you need help from another team.

# Performance, Scalability, and Provisioning

*Keep it concise! Not more than 10 pages before the Appendix.*

How have you designed your system for scale and efficiency? Summarize the conclusions of any performance studies you conducted.

What would be the hardware requirements (based on [redliner](https://iwww.corp.linkedin.com/wiki/cf/display/SOP/Redliner+On+Boarding+Requirements)), [espresso quota](http://go/EspressoQuotaSLO) requirements, kafka production/consumption rate and/or hadoop resource usage? How would your traffic impact other downstream systems?

Please work with your SRE team on this section.

Please link to your service’s SORT page at [go/sort](http://go/sort).

# SLAs, Monitoring, and Alerting

*Keep it concise! Not more than 10 pages before the Appendix.*

Think carefully about metrics that can be used as the basis for SLAs for your service or feature. This is important as it forms the basis for monitoring and alerting. List the set of new metrics you will publish to InGraphs and what they mean. See [Monitoring](http://go/monitoring) for tools available to you. Please work with your SRE team on this section.

Please link to your service’s SORT page at [go/sort](http://go/sort).

# Failure Handling/Graceful Degradation

*Keep it concise! Not more than 10 pages before the Appendix.*

How does your service handle upstream and downstream failures?

# Risks

*Keep it concise! Not more than 10 pages before the Appendix.*

Call out any risks with your proposed design. At LinkedIn, we encourage engineers to take [intelligent risks](http://go/values). List out the possible risks, how you plan to mitigate them, and what data you used to decide to take on these risks.

# Security & Trust

*Keep it concise! Not more than 10 pages before the Appendix.*

What do you need to consider for your service from security, privacy, trust & safety, and legal perspectives? Have you attended Product Security Office Hours? See [Security Assessment Review Process](https://iwww.corp.linkedin.com/wiki/cf/display/ENGS/Security+Assessment+Review+Process) for details.

Remember that [Our Members Come First](http://go/values) in whatever we do.

# Milestones/Rollout

*Keep it concise! Not more than 10 pages before the Appendix.*

Share your milestones and rollout plan along with a link to LiX experiments that may need to be turned on. No need to mention timelines, just the stages in which you plan to build and roll out this design.

Set [measurable and actionable milestones](http://go/values).

# Appendix

Use this section for supporting information or alternative designs primarily to keep your RFC concise and stick to the limit of 10 pages before the Appendix. Move anything that is not critical to your design to this section.

# Q&A

Optionally, you can use this section to summarize frequently asked questions in the comments so that they are documented here for reference.

# References

List any references here.