THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Hidden Integration Costs in Online Controlled Experimentation Platforms

Nick Ross
Director, Data Science Clinic
University of Chicago

Presentation Overview

- Introduction
- Presentation Motivation
- Build vs. Buy
- Some Thoughts and Ramblings
    - Integration Cost Spirals
    - Historical Data / Exporting Data
    - Rabbit Holes
    - Unavailable Features
- Conclusion

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

About me (Academically)

- PhD at UCLA in Management (2012)
- Masters in Economics at UC Davis (2007)
- BA in Applied Math/Statistics at UC Berkeley (2002)
- Research Area(s):
  - Financial Markets
  - Marketing
  - Design of Experiments
  - Applied Big Data ”Stuff”

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

About me (Professionally)

- University of Chicago (2022-Present)
  - Director, Data Science Clinic
  - Instructional Professor
- Director of Backend Engineering & Data Science at The Meta (Not FB, Kovaak) 2020-2022
- Assistant Professor of Data Science at USF (2014-2020)
  - Director of Industry-Academic Partnership
- Director of Analytics at Sega (2014-2015)
- Director of Analytics, Backend Engineering and User Acquisition at TinyCo (2011-2014)
- Senior Consultant Bates White (2002-2006)

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Some Games I've Worked on

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Talk Motivation

- Online Controlled Experiments (”OCE”s) are a powerful tool in product development.
- Deciding how to implement them is a *classic* build-vs-buy decision that can have far reaching and long-term effects on a product.
- Tons of good resources on this decision but it tends to deemphasize or gloss over the *integration* costs.
- This talk will highlight a number of these “hidden” integration costs and how they can bite you in the a**.

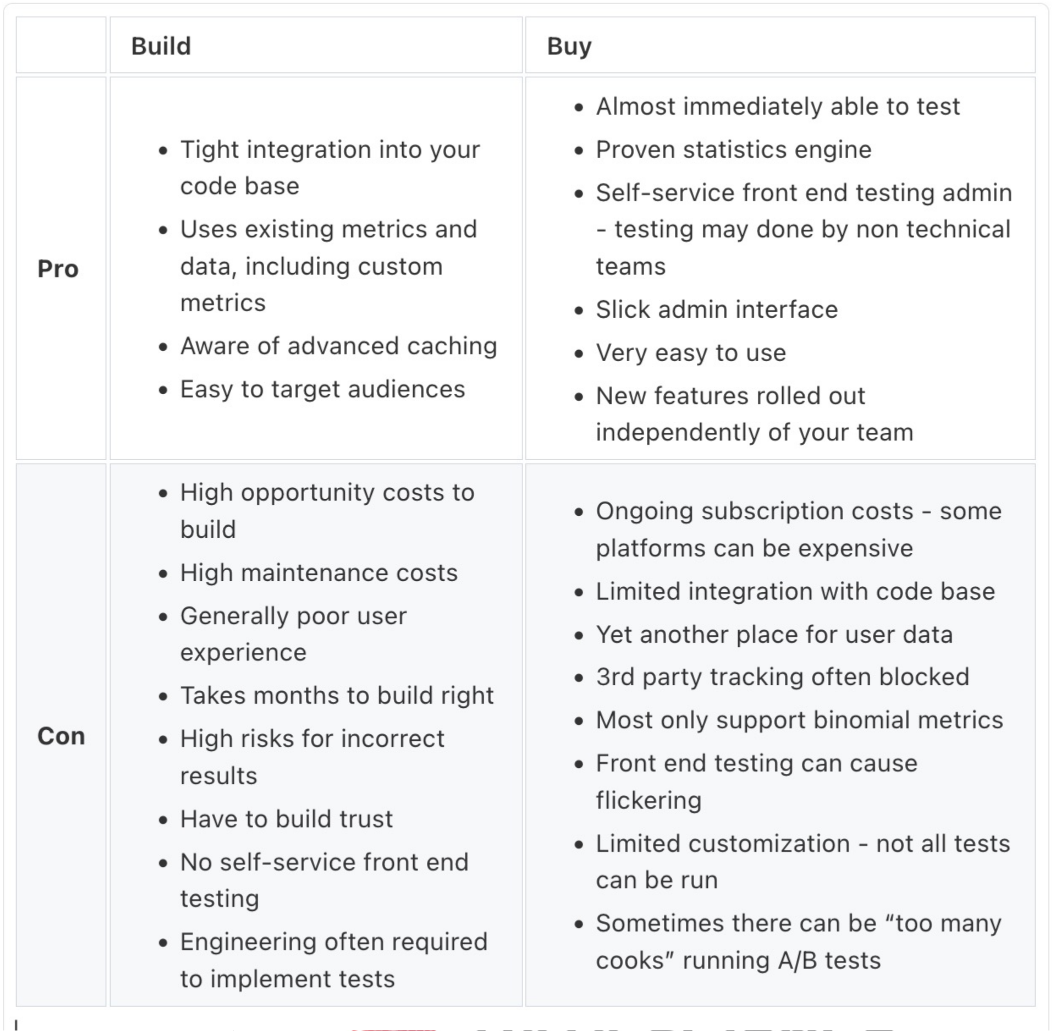
THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Build vs. Buy

- There are a host of 3rd party systems for wading into OCEs:
  - Optimizely
  - KISS Metrics
  - Split
  - Google Optimize
  - Launch Darkly
- The question is should we use one of these or build our own?

Build vs. Buy: Strategic

- Taken from Graham McNicoll's Medium Page, but very representative.
- Lots and lots and lots of version of this can be found.

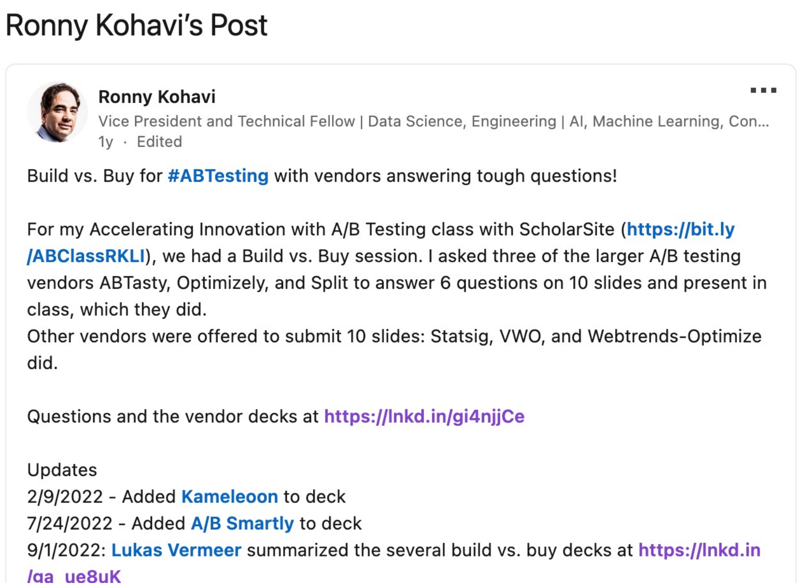


	Build	Buy
Pro	• Tight integration into your code base • Uses existing metrics and data, including custom metrics • Aware of advanced caching • Easy to target audiences	• Almost immediately able to test • Proven statistics engine • Self-service front end testing admin - testing may done by non technical teams • Slick admin interface • Very easy to use • New features rolled out independently of your team
Con	• High opportunity costs to build • High maintenance costs • Generally poor user experience • Takes months to build right • High risks for incorrect results • Have to build trust • No self-service front end testing • Engineering often required to implement tests	• Ongoing subscription costs - some platforms can be expensive • Limited integration with code base • Yet another place for user data • 3rd party tracking often blocked • Most only support binomial metrics • Front end testing can cause flickering • Limited customization - not all tests can be run • Sometimes there can be "too many cooks" running A/B tests

DATA SCIENCE INSTITUTE

Build vs. Buy: Costing

- Ronny Kohavi has a great series of posts and documents around different vendors, their products and pricing.
- Much of the work focuses on specific features of the platform and price (duh).
- Minimize cost given a set of requirements



Ronny Kohavi's Post

Ronny Kohavi
Vice President and Technical Fellow | Data Science, Engineering | AI, Machine Learning, Con...
1y · Edited

Build vs. Buy for **#ABTesting** with vendors answering tough questions!

For my Accelerating Innovation with A/B Testing class with ScholarSite (**https://bit.ly/ABClassRKLI**), we had a Build vs. Buy session. I asked three of the larger A/B testing vendors ABTasty, Optimizely, and Split to answer 6 questions on 10 slides and present in class, which they did.
Other vendors were offered to submit 10 slides: Statsig, VWO, and Webtrends-Optimize did.

Questions and the vendor decks at **https://lnkd.in/gi4njjCe**

Updates
2/9/2022 - Added **Kameleoon** to deck
7/24/2022 - Added **A/B Smartly** to deck
9/1/2022: **Lukas Vermeer** summarized the several build vs. buy decks at **https://lnkd.in/ga_ue8uK**

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Build vs. Buy

- If you read through the build vs. buy literature, you will find a few consistent take-aways:
  - Building costs are way more than you thought
  - Deciding on which 3rd party to go with it a fraught decision and many people get it wrong.

- There is a lot of information, but much of this conversation shies away from the *integration* aspects.
- Integration aspects are company and code base specific.
- So, given that this is already expensive and hard, what do you want to add to the conversation?

Hot Take

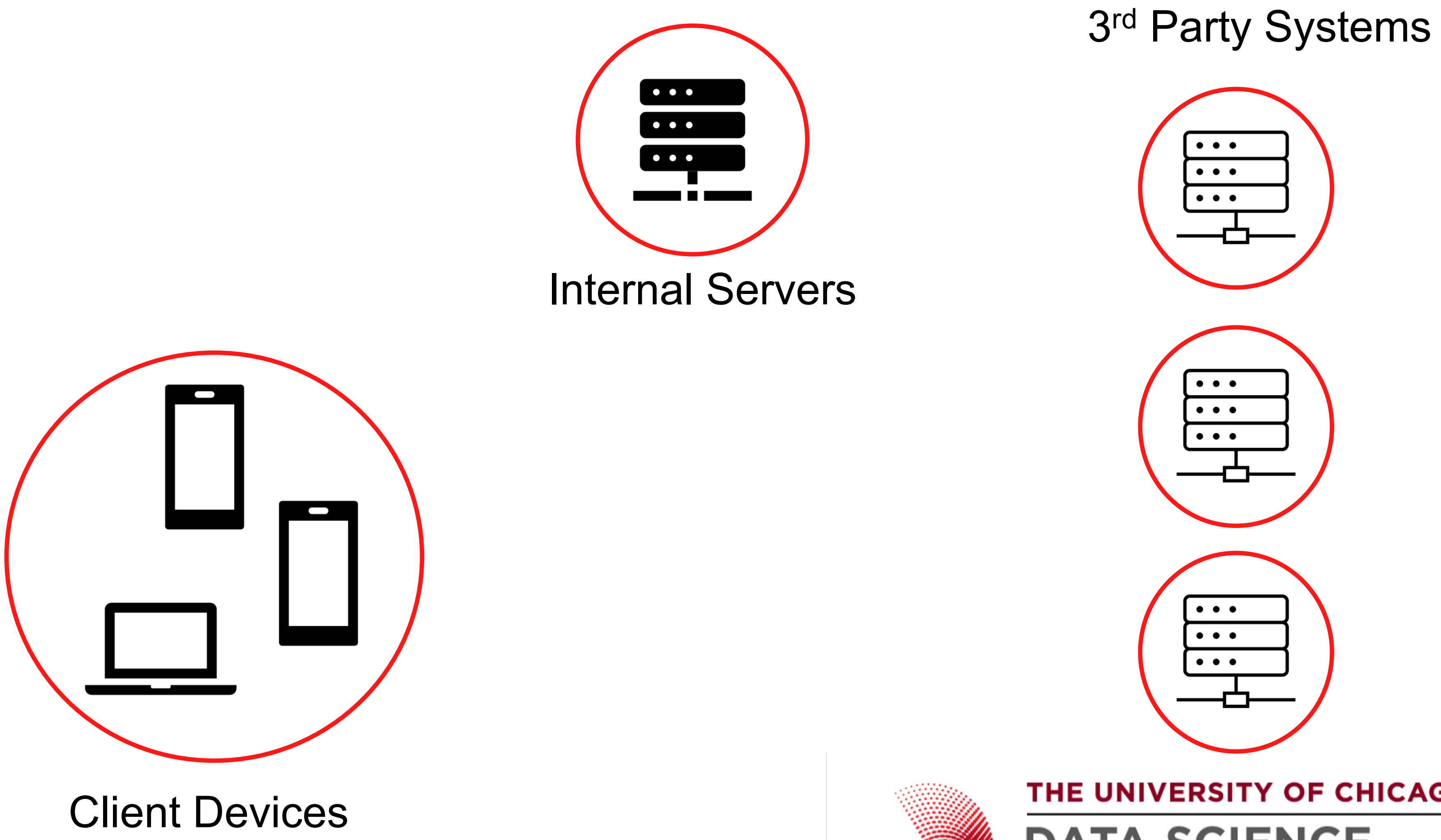
Hidden Integration Costs are much higher than you think and, in many cases, should bias you toward building it yourself.

Why?

- We (almost always) make the build vs. buy decision against the backdrop of our existing tools and infrastructure
- Getting an OCE platform "working" vs. getting an OCE Platform "Integrated"
- *Integrating* and using a suite of 3rd party tools and internal infrastructure is costly.
- Sooo… what do you mean by "Integrating"?
  - To explain this, lets give some more context if we decide to "Buy" an OCE platform

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Typical (simplified) Systems Diagram e-commerce platform



External APIs / 3rd party systems

- Marketing APIs:
  - Push Notifications
  - Emails services
  - CRM Services (HubSpot, Salesforce)
  - Attribution Systems
- ML systems which do overnight batch process
- A/B Testing System
- External Analytics Systems
- Inventory Management System
- Shipping system
- Etc.

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Simple Example: e-commerce platform

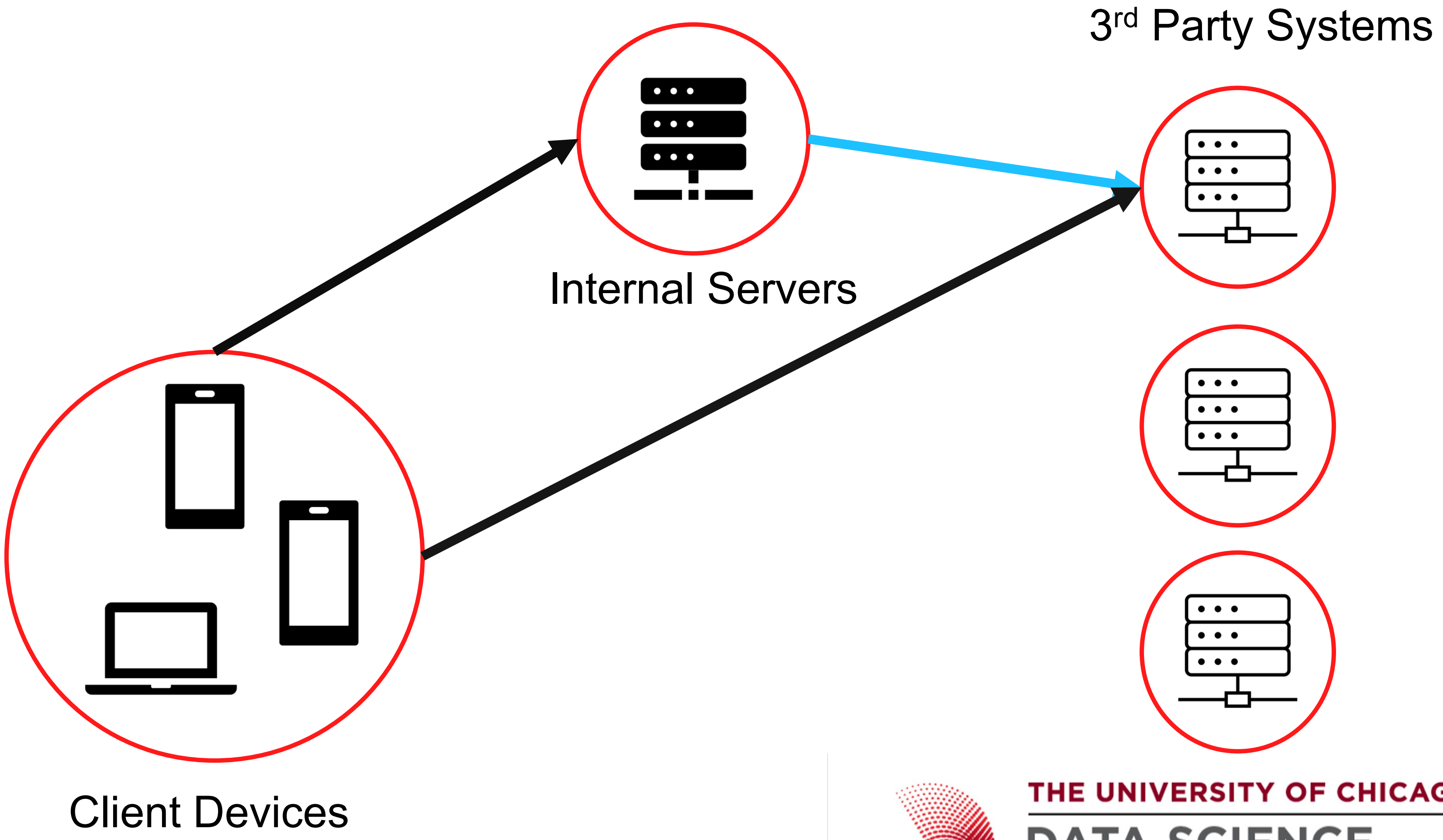
- Our internal server currently keeps track of purchases
- Run an A/B test on a new sales flow and want to know if it increases sales to both “new” customers as well as “existing” customers.
- What information must be sent where?

Simple Example: e-commerce platform

- Our internal server currently keeps track of purchases
- Run an A/B test on a new sales flow and want to know if it increases sales to both “new” customers as well as “existing” customers.
- What information must be sent where?
  - Client <> AB Test system to know which sales flow
  - Internal Server <> AB Test system to know sales status
  - **Historical** Data must be loaded into the AB Test system in order to balance the cohorts and ID new vs. existing sales

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Typical Systems Diagram

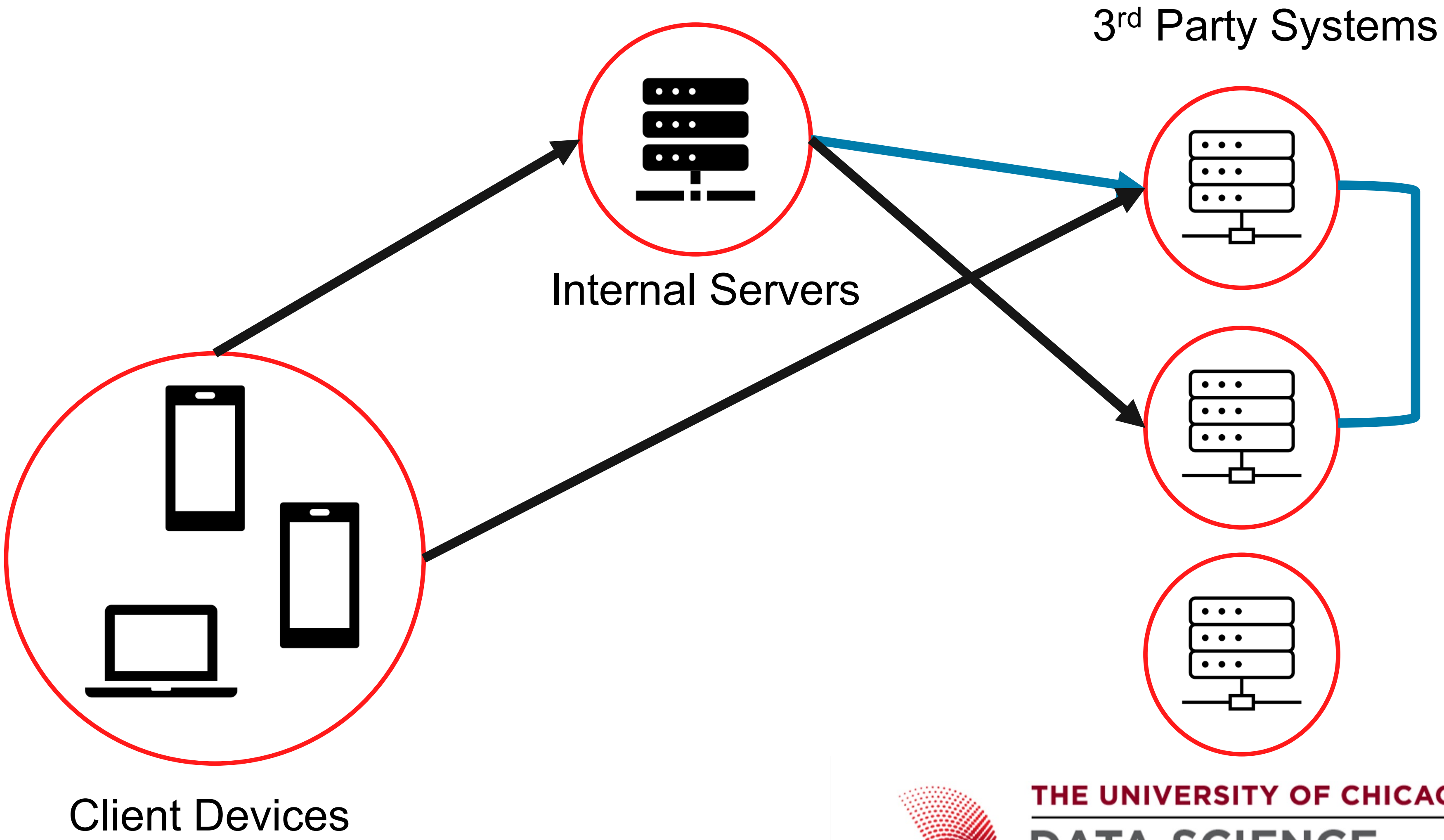


THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Simple Example: e-commerce platform

- I want to run a test on email marketing:
  - If a person puts something in their cart I want to send them an email
  - Test the format of a few subject lines
  - Same customer breakdown: Existing vs. New customers, etc.
- What information must be sent where?
  - Client <> AB Test system to know which sales flow
  - Server <> AB Test system to know when a sales is completed
  - **Historical** Data must be loaded into the AB Test system in order to balance the cohorts and ID new vs. existing sales
  - Marketing <> AB Test
  - Marketing <> Product

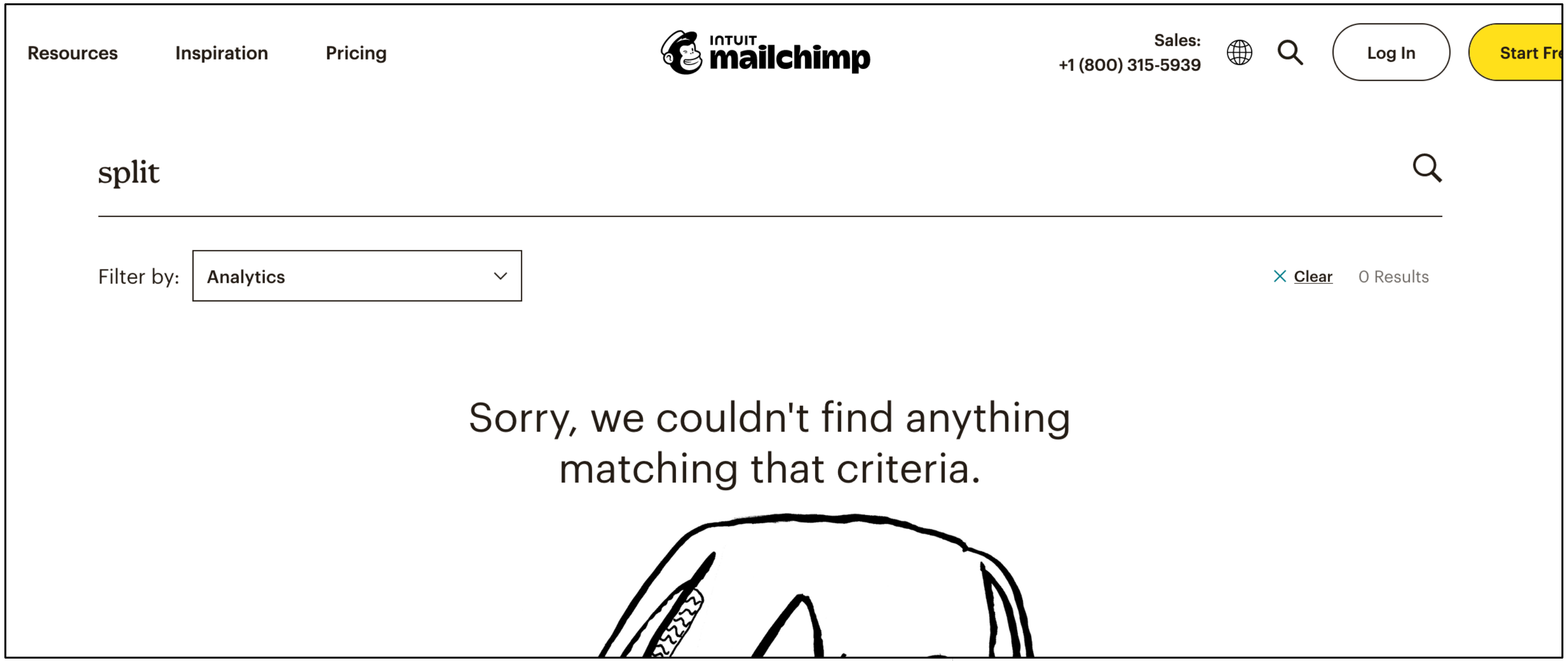
Typical Systems Diagram



THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

WAIT

- How do I connect mailchimp (our 3rd party email marketing solution) to Split.io (our 3rd party OCE platform)?



split

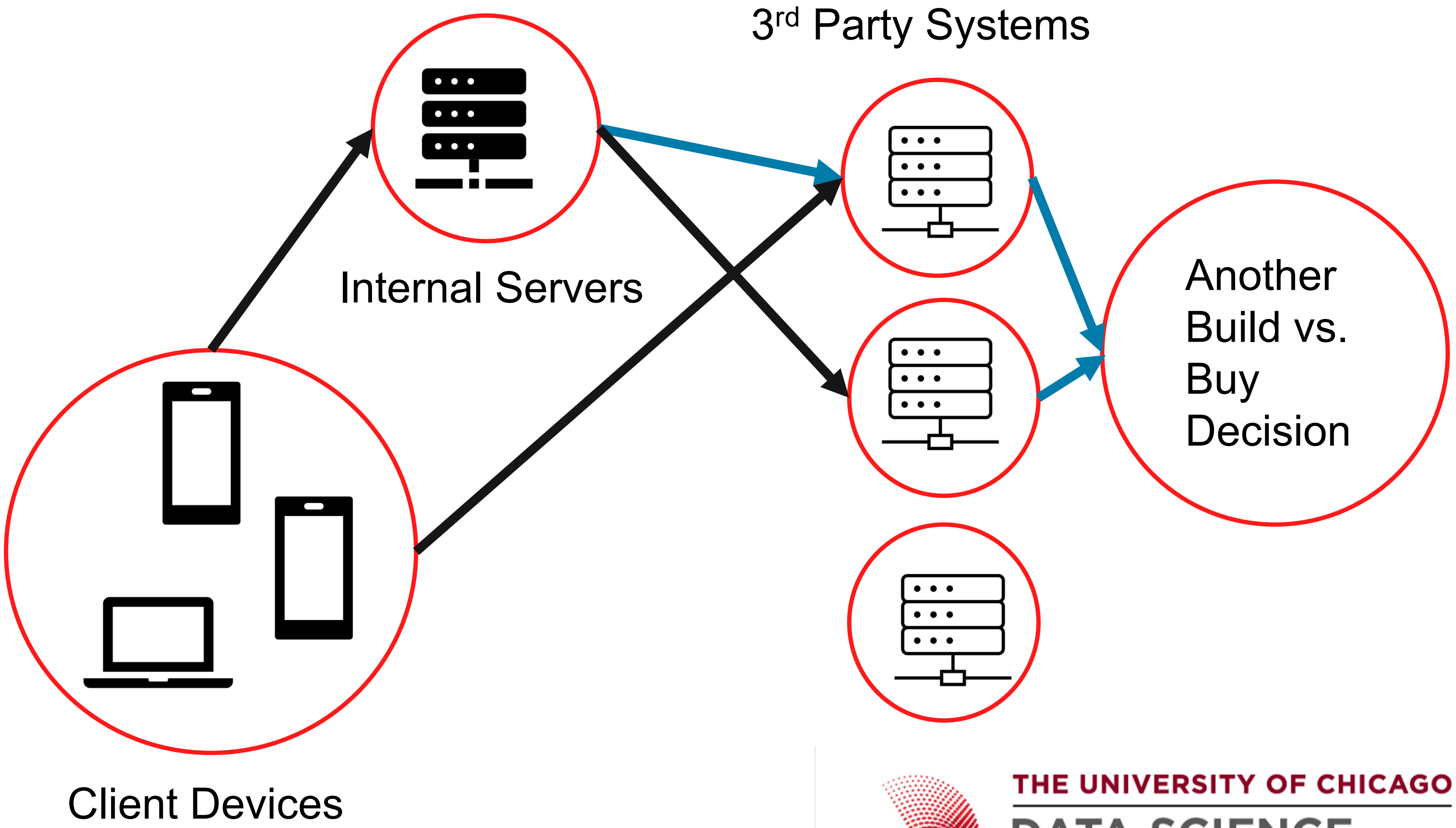
Filter by: Analytics

Clear 0 Results

Sorry, we couldn't find anything matching that criteria.

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Typical Systems Diagram

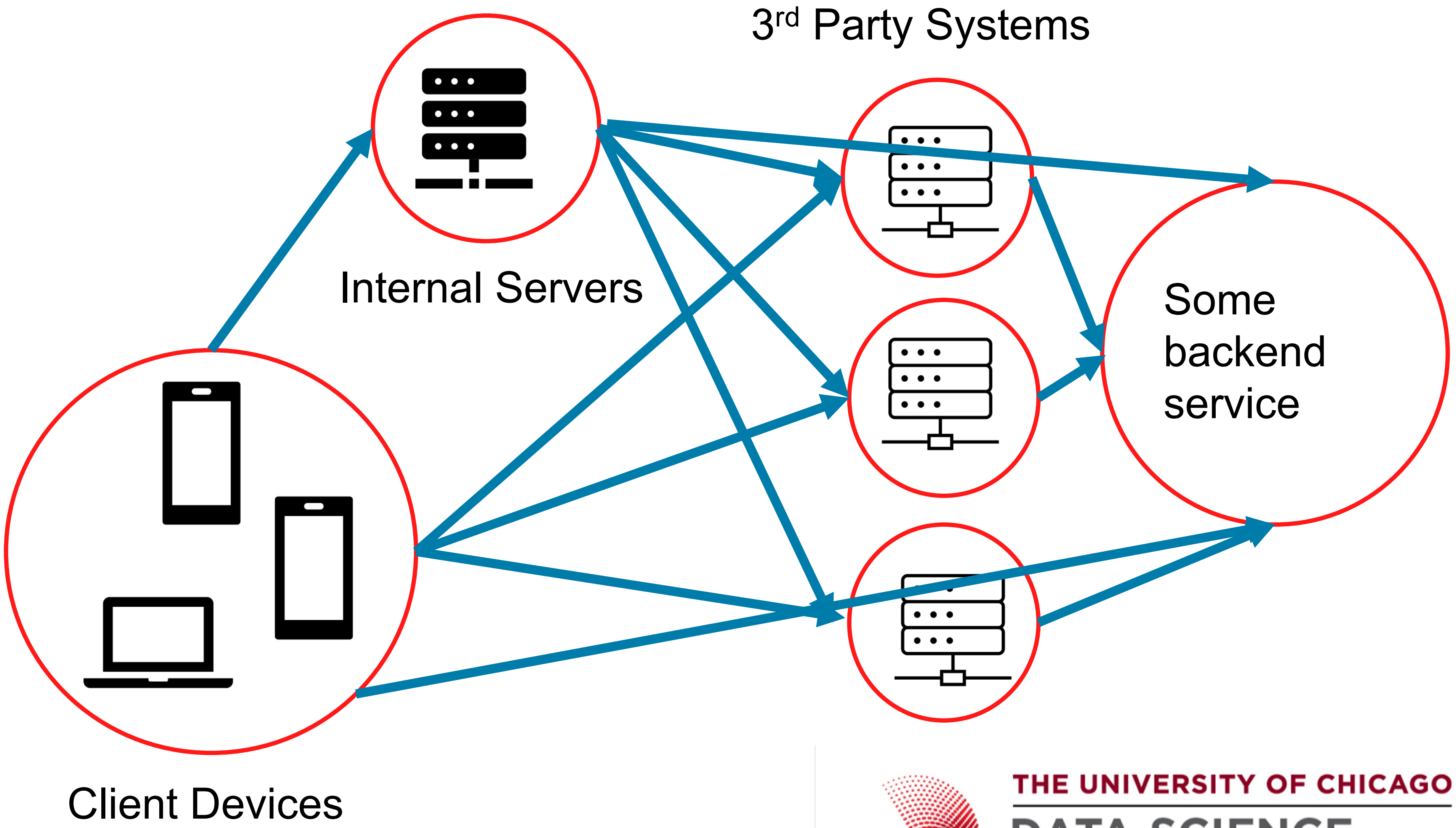


THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

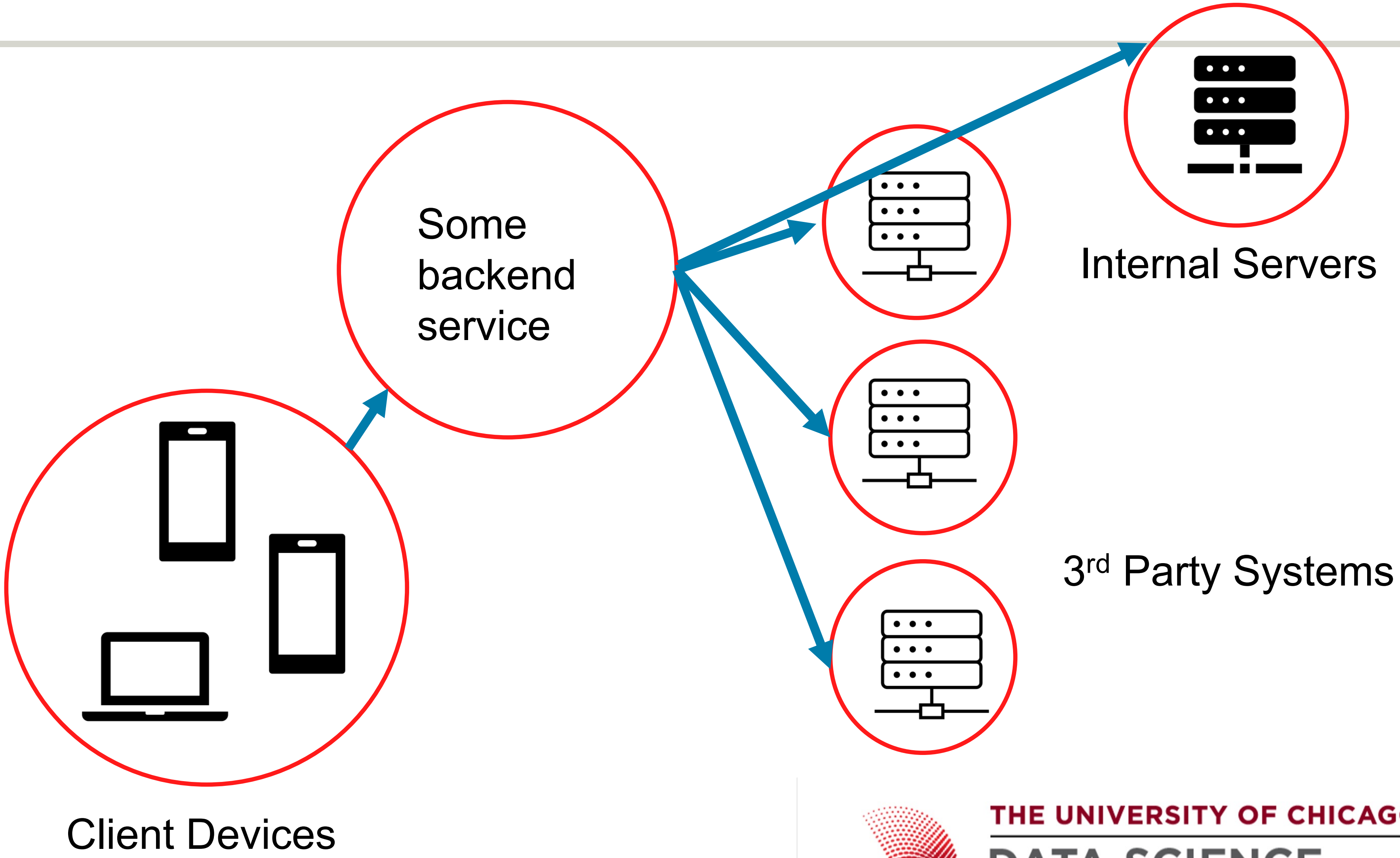
Future

- You can probably guess what will eventually happen.

Typical Systems Diagram



Single Point of Failure Alternative



THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Why are these “hidden” costs?

- Moving from “working” to “fully integrated”
- Getting back to build vs. buy:
  - Discussion centered on working – not on full integration
- Secondary Integrations

Why are these “hidden” costs?

- Every connection in that systems diagram requires:
  - Maintenance
  - Updating as new features are added
  - Testing, logging and alerting systems
  - All of which are engineering time ($$)
- **Every line of code is a liability**
- You can imagine the conversation:
  - *PM*: Now that our AB testing platform is up and running let's run a test on email!
  - *Engineer*: To do that we need to integrate our marketing provider.
  - *PM*: What? I thought we were already running.
  - *Engineer*: Only for front-end design, not for email testing.

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Other hidden costs: Exporting Data

- All major OCE platforms can export data if you want to move to another platform (or just keep the data yourself).
- OCE Platforms don’t like this – it’s a form of vendor lock in.
- Data extraction is usually limited and complex:
  - Ex #1: Requires setting up a special AWS S3 Bucket
  - Ex #2: Creation of special “export” reports which contain subsets of the data.
  - Ex #3: Can only do raw data “going forward”

THE UNIVERSITY OF CHICAGO DATA SCIENCE INSTITUTE

Other hidden costs: Historical Data

- If you want your OCE to be be able to leverage historical data:
  - Different treatments based on specific properties
  - Visualizations, cross-tabs, downstream analysis, etc.
- This historical information needs to be accessible by the OCE:
  - API?
  - Passing the data around (leading to larger payloads)
  - Consistent identification?
- Organizing and providing access requires building additional tooling and glue around your historical data.

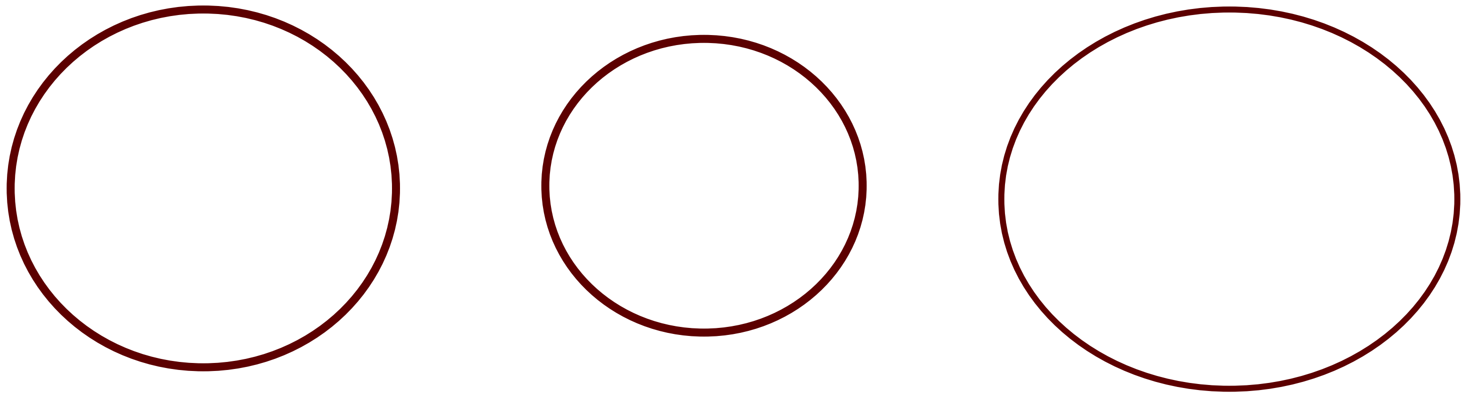
Other hidden costs: Rabbit Holes

- OCEs also provide dashboards to help present information
- The information in these dashboards is often duplicative of information in already existing dashboards
- What happens when they aren't the same?
- Let's say that:
  - OCE Dashboard says there were 101 sales yesterday
  - Internal dashboard says 99 sales yesterday
- "Hidden" Costs:
  - Reputational
  - Investigation Costs
- **Every dashboard is a liability**

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Other hidden costs: Unavailable Features

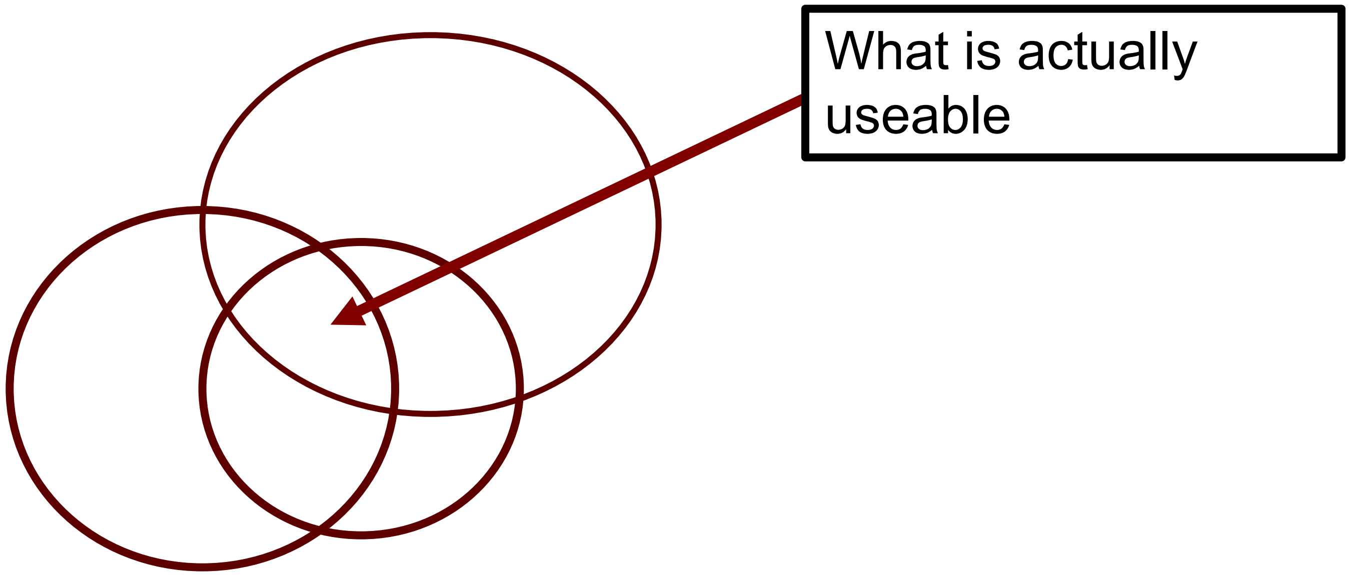
- OCE platforms have features
- Marketing platform have features
- Internal systems have features
- What is available to the end user?



THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

Other hidden costs: Unavailable Features

- OCE platforms have features
- Marketing platform have features
- Internal systems have features
- What is available to the end user?



Conclusion

- Most of the build vs. buy discussion shies away from talking about integration costs
  - Company, tech stack and feature-use specific
  - Hard to quantify
- Moving from getting the platform "working" to getting the platform "integrated" is (IMO):
  - Much higher cost than getting an OCE platform "working"
- Specific Costs:
  - Secondary Integrations
  - Unavailable Features
  - Rabbit Holes
  - Historical Data
  - Data Exports

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE