

# Finance Systems

Build vs. Buy:  
How to Make the Right Decision

Leapfin



SEAT  
GEEK

January 18, 2021

# Housekeeping



You are muted for the duration of the webinar



Please submit questions via the Q&A module or reach out to [marketing@leapfin.com](mailto:marketing@leapfin.com)



The session is being recorded and will be shared afterwards

## Moderator



**.L Leapfin**

**Raymond Lau**  
CEO

## Speaker



**SEAT  
GEEK**

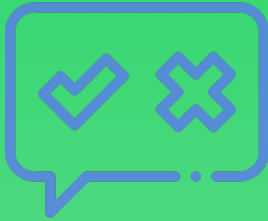
**Joe Blanchett**  
Sr Product Manager,  
Finance Systems

# AGENDA

---

- **Presentation (20 mins)**
  - What should your unified financial backend system look like?
  - What are the different components and considerations?
- **Case study: SeatGeek (20 mins)**
  - How SeatGeek evaluated the build vs buy scenario
- **Q&A**

# Poll Question



**What is the most important part of evaluating a build vs buy scenario for finance systems?**

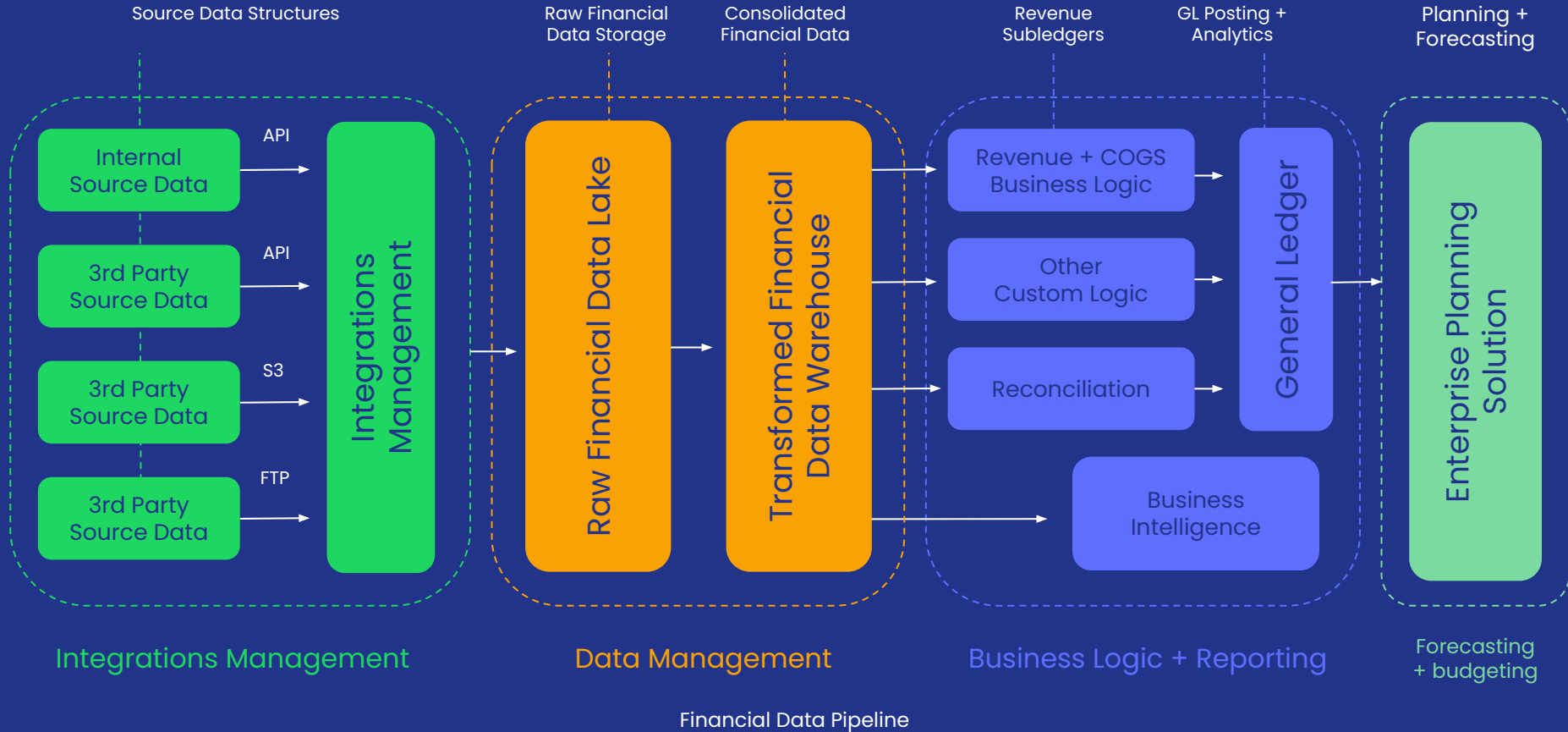
- a) Deciding on evaluation criteria
- b) Coming up with a scorecard
- c) Projecting ROI and business case
- d) All of the above

# IPO-bound companies have a lot of finance requirements

- IPO-readiness and SOX compliance
- Eliminate reliance on manual, error-prone processes
- Audit-readiness
- Company growth: new geographies, new revenue streams



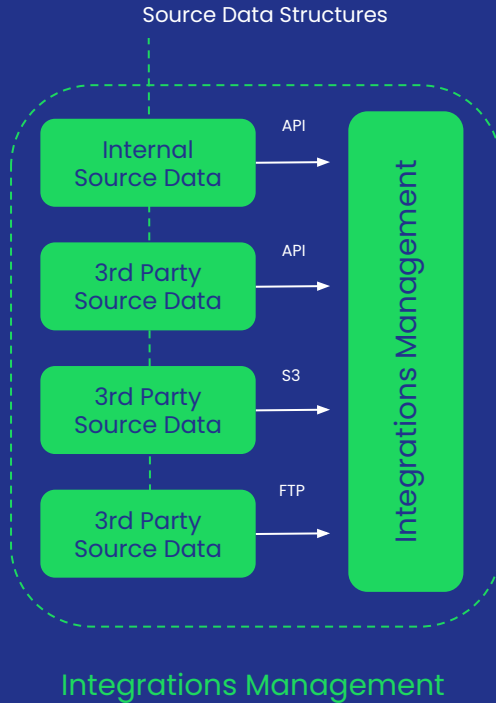
# Key components of a unified financial backend system



**What should you be evaluating when building or buying a finance system?**



# Key component #1: Integrations management



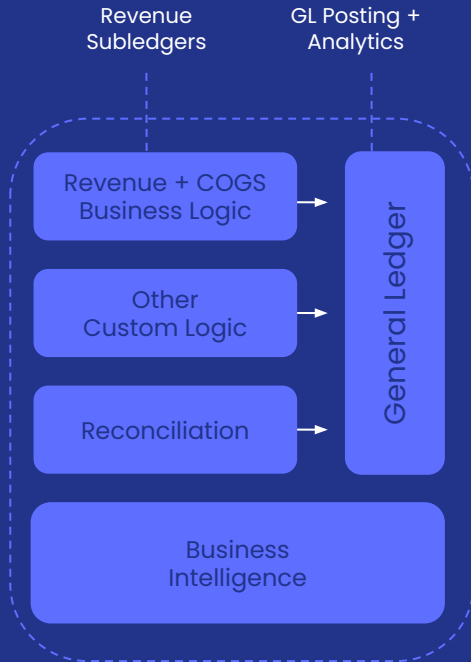
Sub-components	Considerations
<b>1</b> Data Connectors	<ul style="list-style-type: none"><li>• Introducing new payment processors as the business expands into new geographies or business models</li><li>• Updated API versions</li><li>• Optimizing data ingestion for high transaction volume</li></ul>
<b>2</b> Scheduling	<ul style="list-style-type: none"><li>• Data dependencies should be considered to prevent race conditions and/or data integrity issues</li><li>• Data providers may delay publishing data, which would require active monitoring and ad-hoc data backfills</li></ul>

# Key component #2: Data management



Sub-components	Considerations
1 Consolidated Financial Data Warehouse	<ul style="list-style-type: none"><li>• Each data source will have its own data schema and must be normalized into a single data schema before data can be processed</li><li>• Security compliance and controls is critical given the sensitivity of financial data</li></ul>

# Key component #3: Business logic + reporting

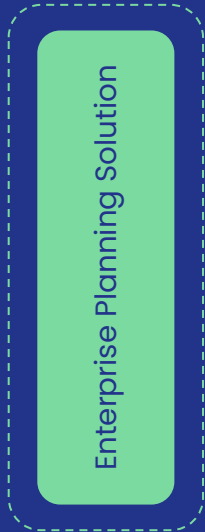


Business Logic + Reporting

Sub-components	Considerations
<b>1</b> Revenue Subledger	<ul style="list-style-type: none"><li>• Revenue numbers must be accurate and timely and reliable</li><li>• The order of data processing and journal entry creation is critical; however, data received from source systems may be out of order.</li><li>• New accounting standards (e.g. ASC 605-606)</li><li>• Differential update as source data may only store the latest state of data</li><li>• User access control/change management (sox compliance)</li><li>• UI/Reporting to ensure finance has the reports they need - accountants should not be running SQL queries</li></ul>
<b>2</b> General Ledger Posting	<ul style="list-style-type: none"><li>• Accounting needs the ability to close or "lock" a period. Any programmatic changes that may impact historical periods need to be properly captured in the current period. Closed or "locked" periods cannot be altered.</li></ul>

# Key component #4: Business logic + forecasting

Planning +  
Forecasting



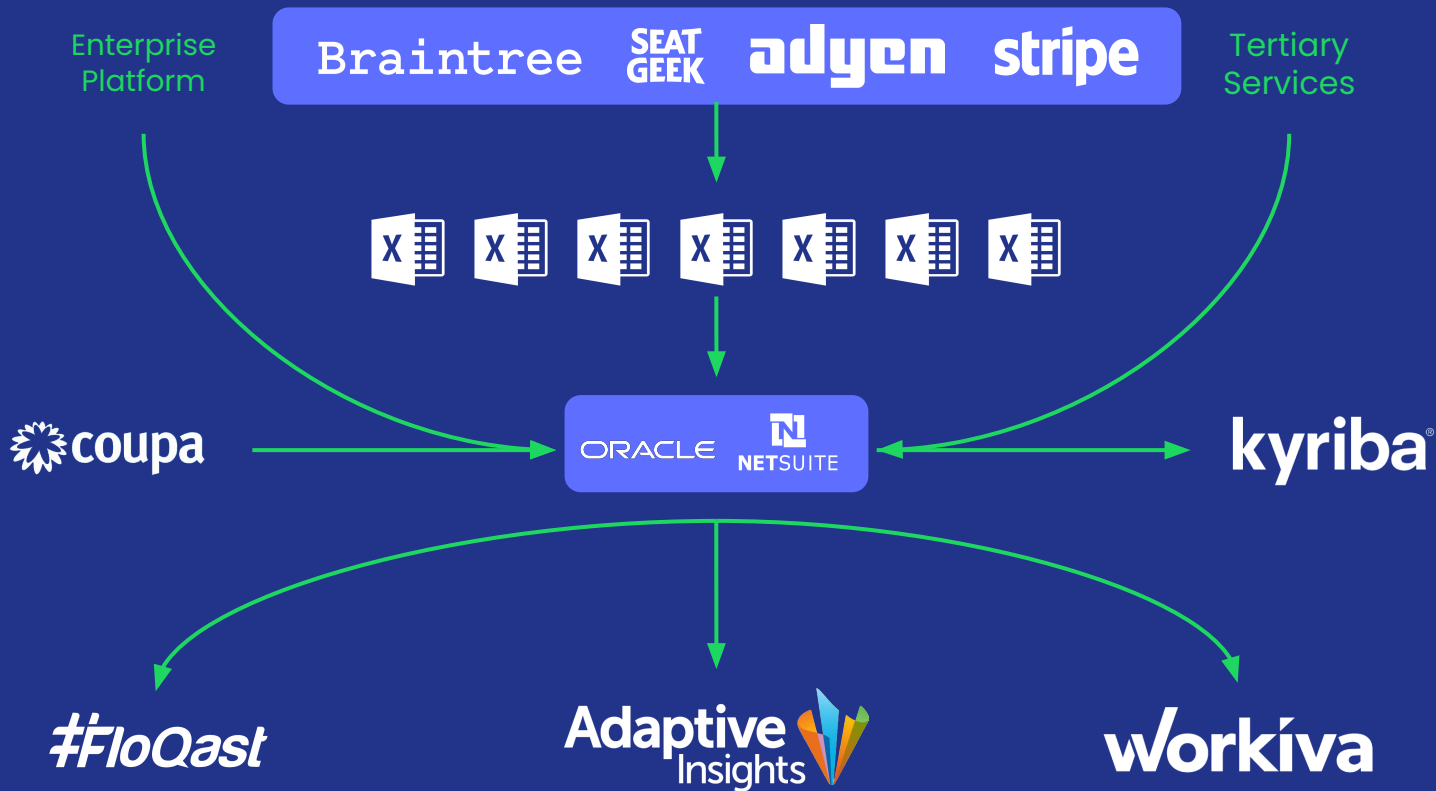
Forecasting  
+ budgeting

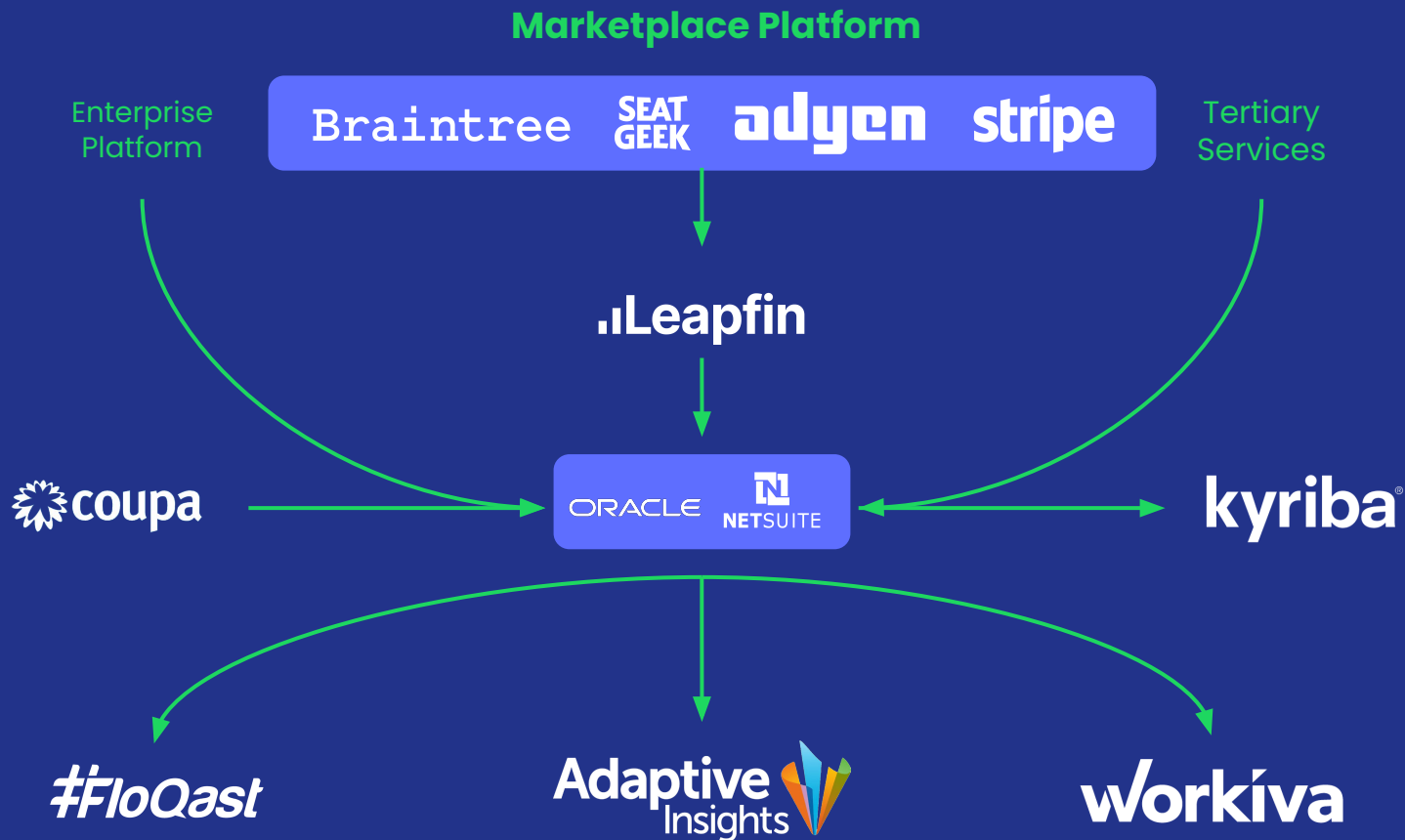
Sub-components	Considerations
<b>1</b> Enterprise Planning Solution	<ul style="list-style-type: none"><li>• More frequent GL posting is better</li><li>• Good integration with planning system to get numbers faster; can refresh forecast click of a button</li></ul>

# Case Study

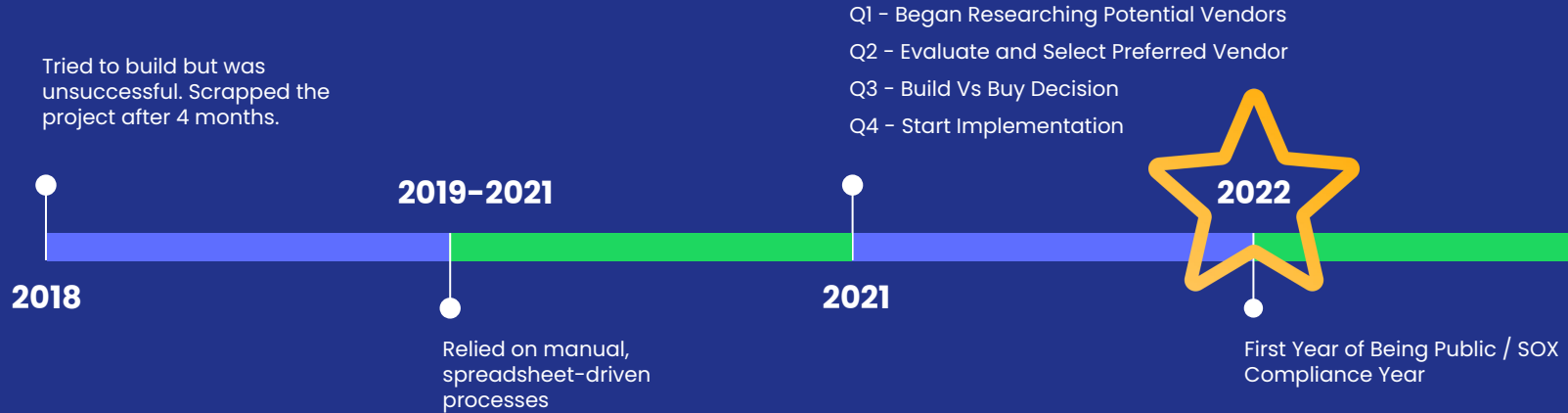
How **SeatGeek** successfully  
evaluated **build vs buy**

### Marketplace Platform





# Build vs Buy Timeline





# What did we consider?

1

**Direct Cost (People)**

2

**Opportunity Cost**

3

**Time to Value**

6 months longer to build in-house with less functionality and UX

4

**Future-proofing considerations**

New data sources and payment processors, new lines of business

# Sample Build vs. Buy Scorecard

Attribute Category	Attribute	Buy Score	Build Score	Basis for Score
<b>Technical</b>	API Based Integrations			
	Differential Update			
	Integrations with PSPs			
	Able to Post to NetSuite			
<b>Functional</b>	Easy-to-Use Reporting Interface			
	Can Easily Pull Audit Samples			
	Revenue by Class Report			
	Waterfall Reports for Accrual Based Accounts			
	Transaction Audit Trail			
	Configurability / Flexibility			
	View Impact Across Periods			
<b>Compliance</b>	SOC Report			
	Ability to Audit Access			
	Ability to Audit Config			

# Q&A



Thank You

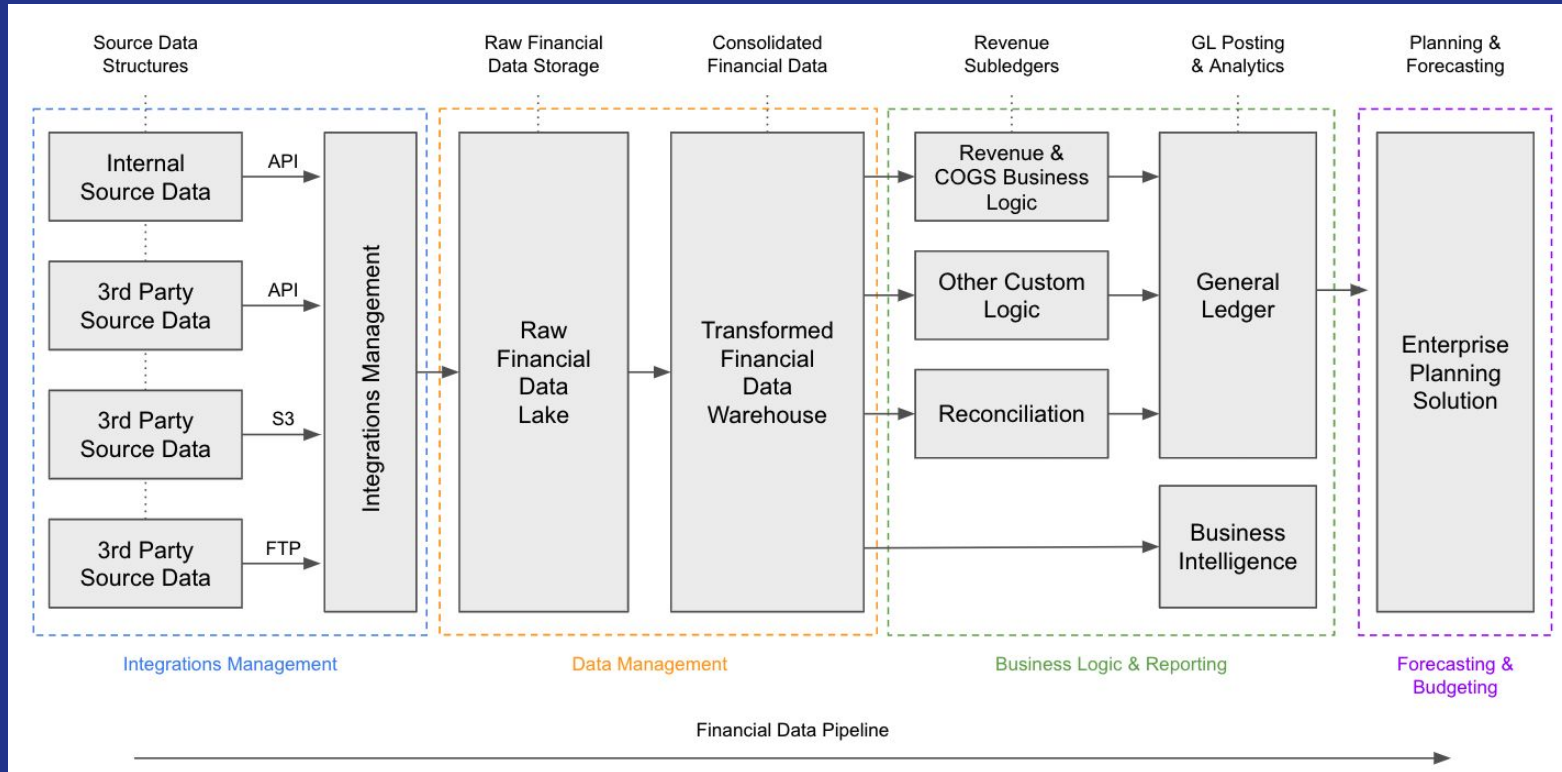


Leapfin Revenue Platform helps

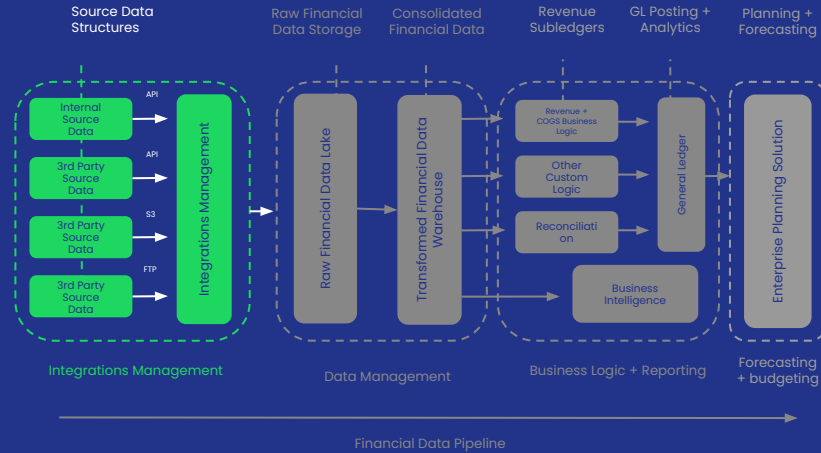
*If you have any questions regarding this content, feel free to reach out to [marketing@leapfin.com](mailto:marketing@leapfin.com)*



# Key components of a unified financial backend system



# Key component #1: Integrations management



Sub-components		Considerations
1	Data Connectors	<ul style="list-style-type: none"> <li>Introducing new payment processors as the business expands into new geographies or business models</li> <li>Updated API versions</li> <li>Optimizing data ingestion for high transaction volume</li> </ul>
2	Scheduling	<ul style="list-style-type: none"> <li>Data dependencies should be considered to prevent race conditions and/or data integrity issues</li> <li>Data providers may delay publishing data, which would require active monitoring and ad-hoc data backfills</li> </ul>