

Q4 2019



The Data Platform for your  
Software Development Life Cycle

7 out of 10  
enterprise IT initiatives fail

# IT projects fail because executives have no visibility

source{d}



## Lack of visibility comes from:

Source code, software development and business data are spread across many silos

Large volume, variety, intricacy and versions of data

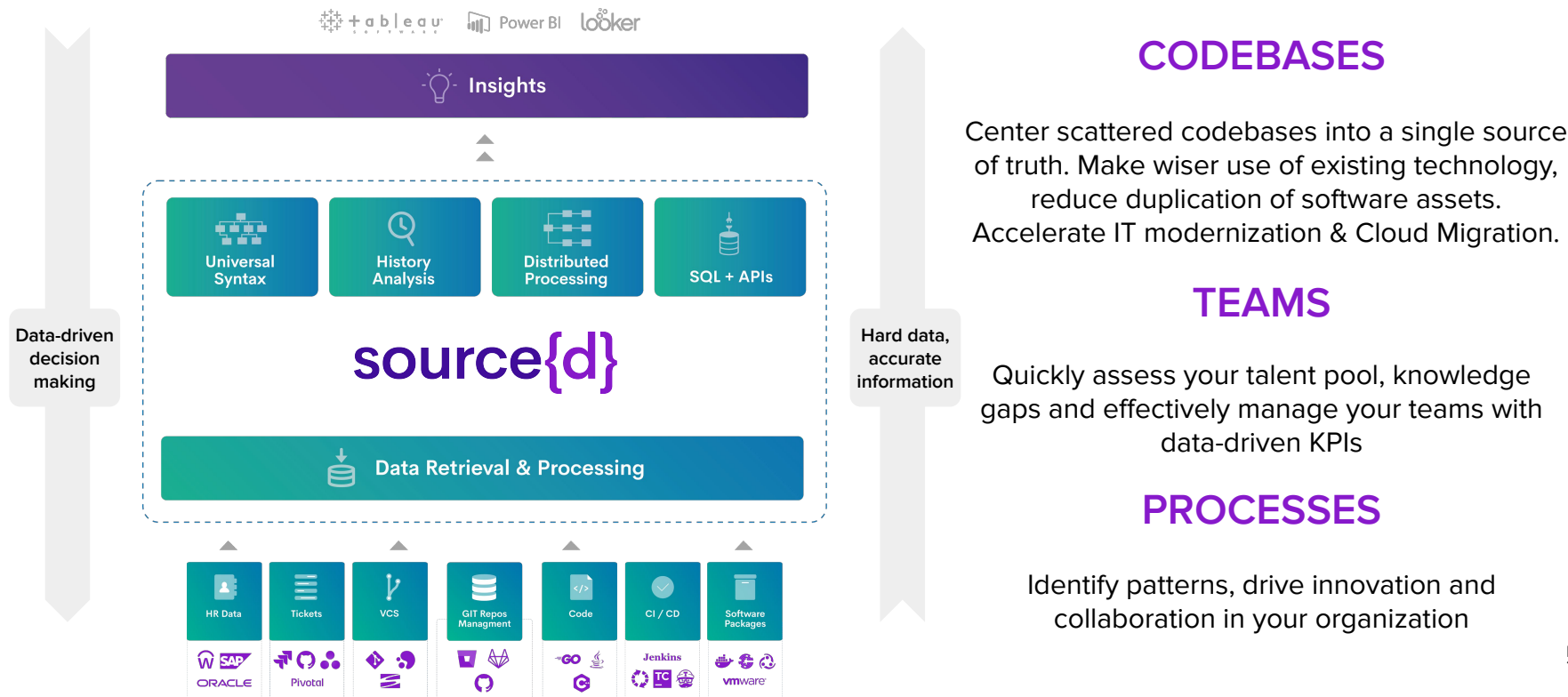
Retrieving, storing and querying at scale is hard

Shadow IT and “Dark Data” due to developer frustration with legacy systems & slow processes

source{d}  
gives IT executives visibility into  
codebases, IT teams & processes

# From “dark data” to enlightened decisions

source{d}



## CODEBASES

Center scattered codebases into a single source of truth. Make wiser use of existing technology, reduce duplication of software assets. Accelerate IT modernization & Cloud Migration.

## TEAMS

Quickly assess your talent pool, knowledge gaps and effectively manage your teams with data-driven KPIs

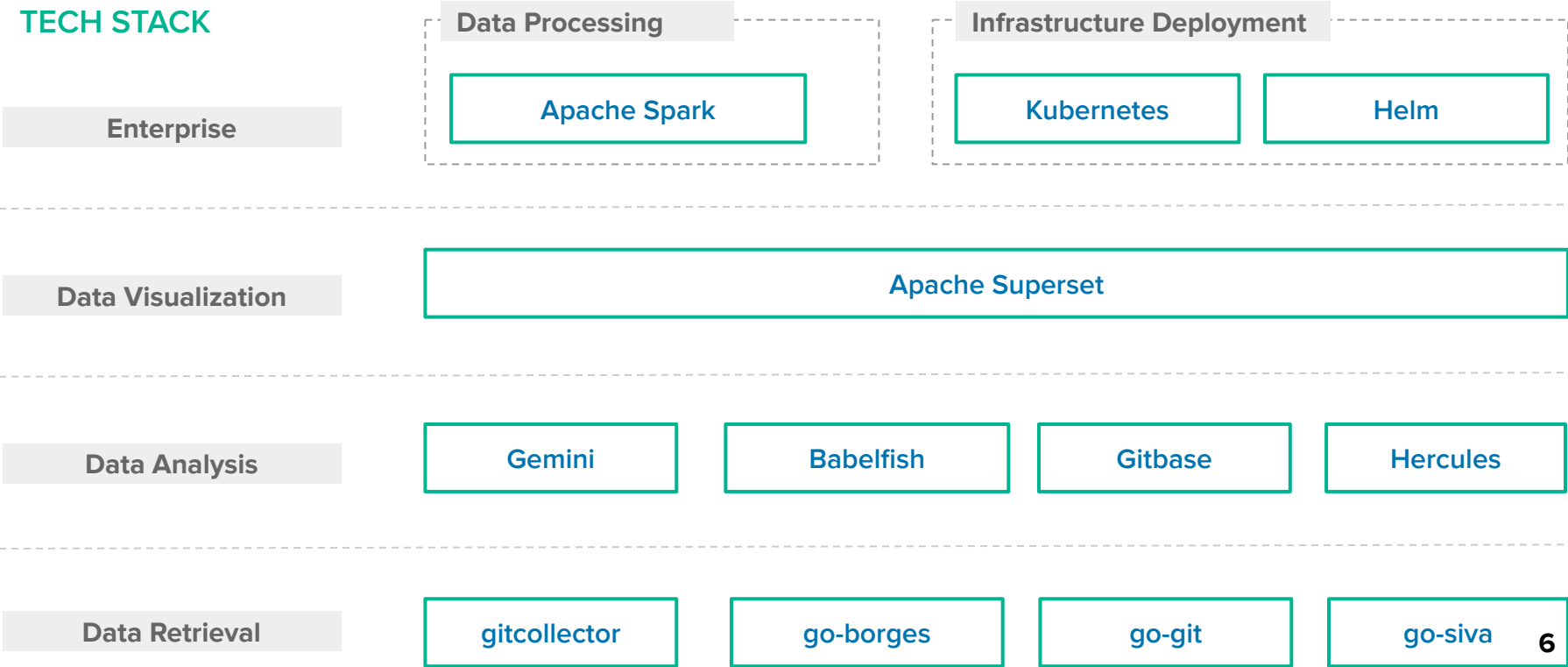
## PROCESSES

Identify patterns, drive innovation and collaboration in your organization

# Component overview

source{d}

## TECH STACK



# Powerful analyses on demand

source{d}

## ACCESSIBLE

Quickly answer your most complex questions  
Empower your teams through SQL queries

The screenshot shows the source{d} SQL Lab interface. On the left, there's a sidebar with 'Database: mysql', 'Schema: gitbase', and a list of tables including 'blobs', 'commit\_blobs', 'repository\_id', 'commit\_hash', and 'blob\_hash'. The main area has a query editor with a SQL query for analyzing commit data. A red arrow points to the query with the text 'write questions'. Below the editor is a 'Results' section showing a table with columns: lang, code, comments, blanks, and files. The table contains data for various languages like Go, Markdown, XML, JSON, HTML, and YAML. A green arrow at the bottom left points to the results with the text 'get answers'.

lang	code	comments	blanks	files
Go	7445891	0	638668	18654
Markdown	63927	0	24337	694
XML	24412	0	934	30
JSON	23245	0	0	223
HTML	14597	0	2390	116
YAML	6082	0	792	231

## LANGUAGE AGNOSTIC

Universal ASTs\* enable deep & wide analyses across  
programming languages through semantic concepts

The screenshot shows the source{d} Web interface. On the left, there's a 'Web' tab with a 'LANGUAGE' dropdown set to 'AUTO (JAVA)'. Below it is a code editor showing Java source code for a 'Hello' servlet. A red arrow points from the 'service' method call in the code to the 'UASTs' section on the right. The 'UASTs' section shows a tree structure representing the semantic analysis of the code, with nodes like 'Function', 'Name', and 'Identifier'. The 'Name' node is highlighted in yellow and labeled 'service'.

```
1 // hello.java
2 import java.io.*;
3 import javax.servlet.*;
4
5 public class Hello extends GenericServlet {
6     public void service(final ServletRequest request, f
7         throws ServletException, IOException {
8         response.setContentType("text/html");
9         final PrintWriter pw = response.getWriter();
10        try {
11            pw.println("Hello, world!");
12        } finally {
13            pw.close();
14        }
15    }
16 }
17
```

UASTs

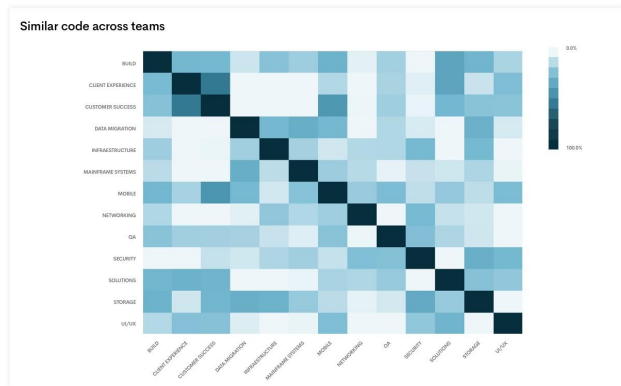
- constructor: 'false'
- extraDimensions2: null
- javadoc: null
- modifiers: []node
  - + Node
- name
  - @role: []string
  - 'Function'
  - 'Name'
  - @type: 'uast:Identifier'
  - Name: 'service'
- parameters: []node

\* Universal ASTs were developed by source{d} as a language agnostic layer on top of source code, analyze source code independent of the diversity of programming languages

# Machine Learning on Source Code

source{d}

- **Expertise Assessment:** measuring Topic Modelling, Code Ownership, Identifier embeddings, etc
- **Similar & Duplicate Code Analysis:** extracting the natural language elements in code (how developers choose to name entities, for example) and the structure of the code
- **Collaboration Assessment:** Exploring the topological structure of commit activity, usage of programming languages and topics of source code identifiers
- **Identity matching:** building connected components and applying clustering techniques
- **Code Representation:** researching the best ways to encode code to have a strong foundation on which to build further products.

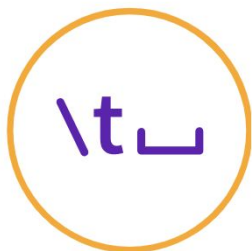




# Solving key enterprise challenges for IT execs

source{d}

**IT Modernization &  
Compliance**



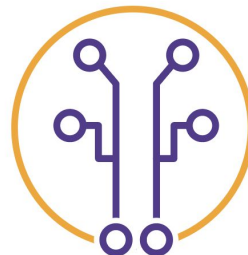
**Chief Architects, VP of IT  
Applications**

**Cloud Native &  
DevOps  
Transformation**



**Chief Architects, Head of  
Continuous Delivery**

**Engineering Effectiveness  
& Efficiency**



**Head of Developer Experience,  
VP of IT Operations**

**Talent Assessment  
& Management**



**VP and Director of  
Engineering**