DATA USE ASSESSMENT REPORT

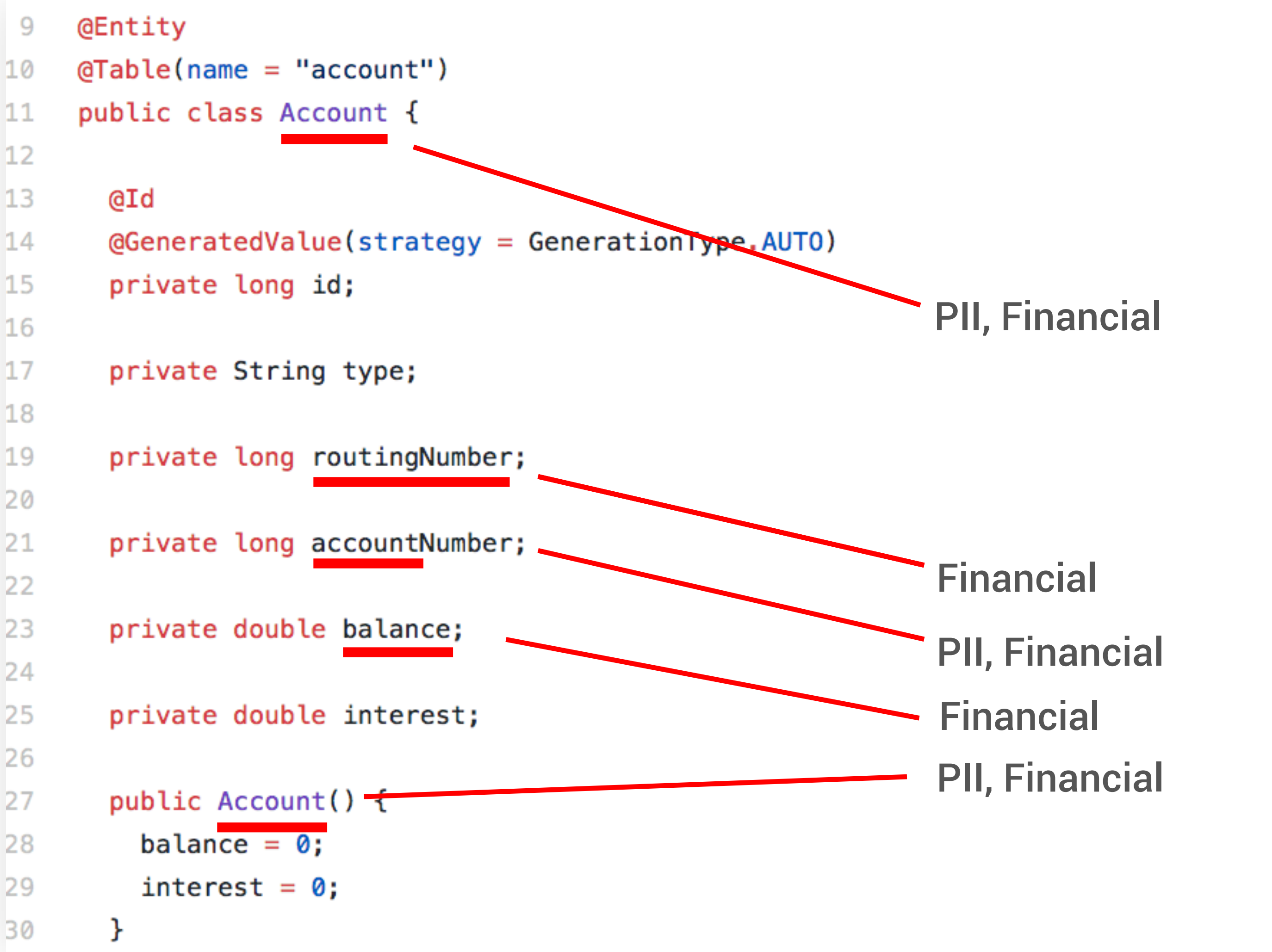
UNDERSTAND THE COMPLETE FOOTPRINT OF YOUR SENSITIVE DATA AS IT PARTICIPATES IN YOUR BUSINESS LOGIC

HOW DATA IS CLASSIFIED AND ANALYZED

Using ShiftLeft's core code analysis technology, we can provide a complete picture of what data exists within your applications and surrounding infrastructure, as well articulate how this data flows through application logic, referencing exact lines of code (should deeper investigation be needed).

IT'S ALL ABOUT DATA FLOWS

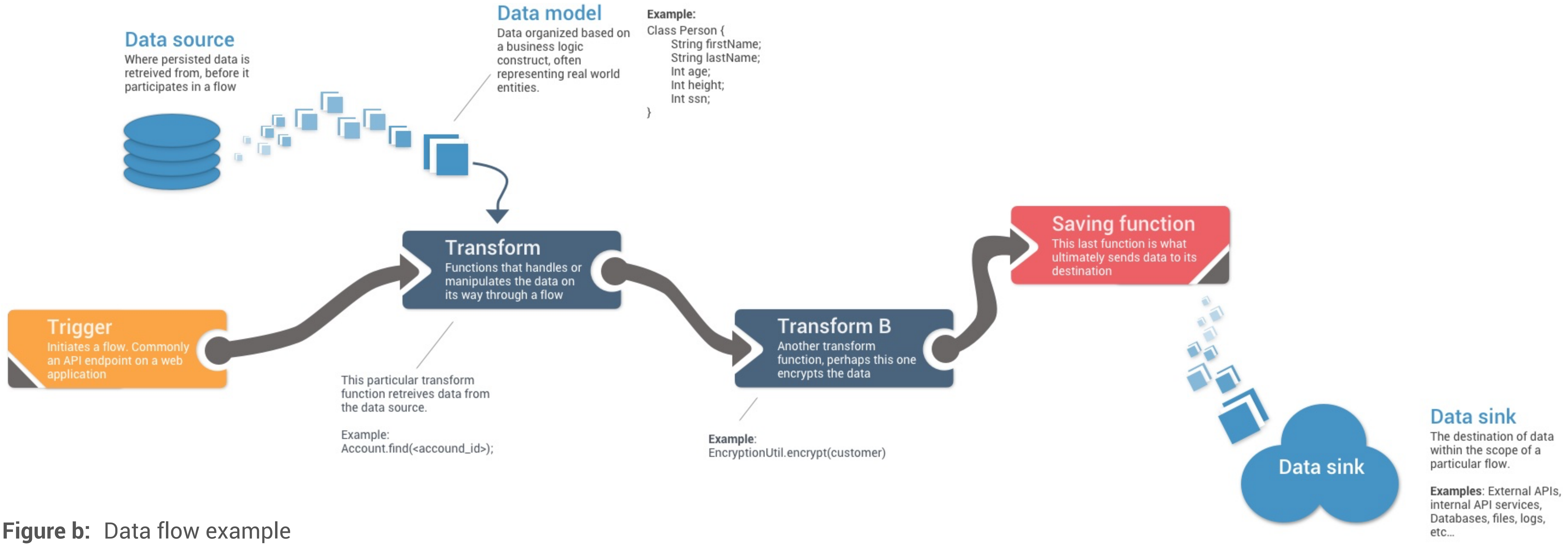
Our analysis uses two approaches to understand what data exists, and how that data propogates through application code. First, we look through source code using a natural language processor, and identify and classify variables that are likely sensitive. **Figure a**. show's how for a given class definition, we will recognize keywords and associate them with common data catagories:



```
@Entity
@Table(name = "account")
public class Account {

  @Id
  @GeneratedValue(strategy = GenerationType.AUTO)
  private long id;

  private String type;

  private long routingNumber;

  private long accountNumber;

  private double balance;

  private double interest;

  public Account() {
    balance = 0;
    interest = 0;
  }
```

Figure a: Data recognition and classification

Next, using a graph analysis approach, we track how data participates in data flows throughout the application. The following diagram (Figure b) depicts how we describe a common data flow



Data source
Where persisted data is retreived from, before it participates in a flow
Data model
Data organized based on a business logic construct, often representing real world entities.
Example:
Class Person {
String firstName;
String lastName;
Int age;
Int height;
Int ssn;
}
Trigger
Initiates a flow. Commonly an API endpoint on a web application
Transform
Functions that handles or manipulates the data on its way through a flow
This particular transform function retreives data from the data source.
Example:
Account.find(<accound_id>);
Transform B
Another transform function, perhaps this one encrypts the data
Example:
EncryptionUtil.encrypt(customer)
Saving function
This last function is what ultimately sends data to its destination
Data sink
Data sink
The destination of data within the scope of a particular flow.
Examples: External APIs, internal API services, Databases, files, logs, etc...

Figure b: Data flow example

WHY USE A SHIFTLEFT REPORT?

- **GDPR and similar legal requiremens**
  Laws and regulations on the storage and use of personal data are changing rapidly. ShiftLeft's product solution and reports offer a way to continously and automatically stay up to date.

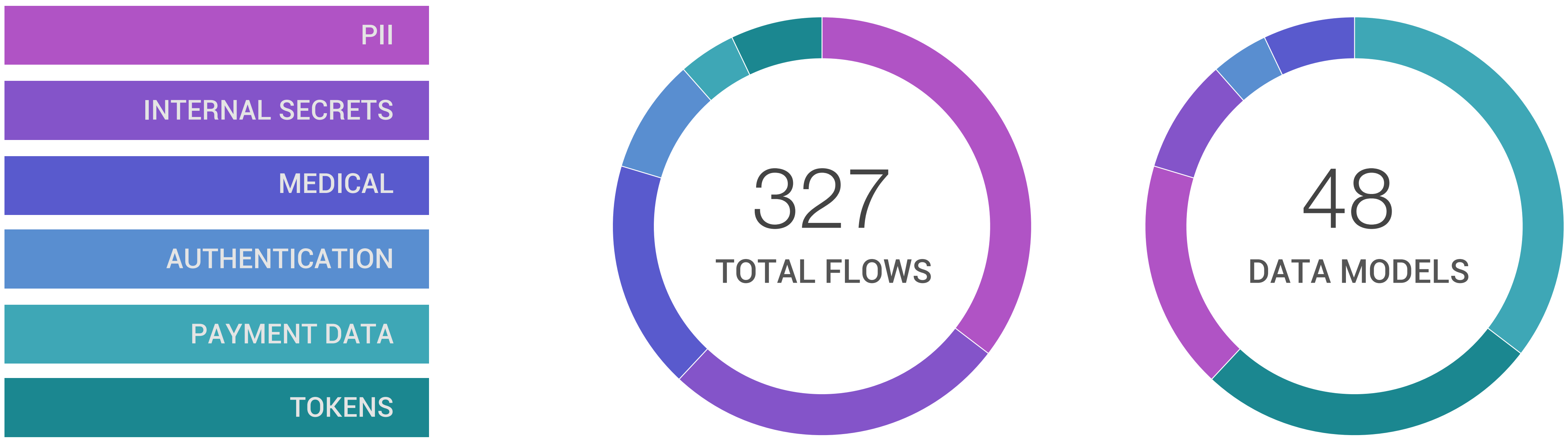
- **Data use compliance**
  Depending on the type of data and business, there are a variety of compliance efforts that are typically needed to assure security standards are being upheld.

- **Ongoing security strengthening**
  Regardless of what laws or compliance areas affect your business, customer data is the lifeblood and currency of modern software business. Contiuous understanding of how that data is being used or exposed is necessary to protect and grow your company.

APPLICATION INFO

APPLICATION NAME	io.shiftleft-hello-shiftleft-jar
PROJECT URI	git@github.com:ShiftLeftSecurity/HelloShiftLeft.git
BRANCH/REF	heads/master
COMMIT ID	26409bf537028669a3d156adce08d0e6a6252fbb
COMMIT AUTHOR	Chetan Conikee <conikee@gmail.com>

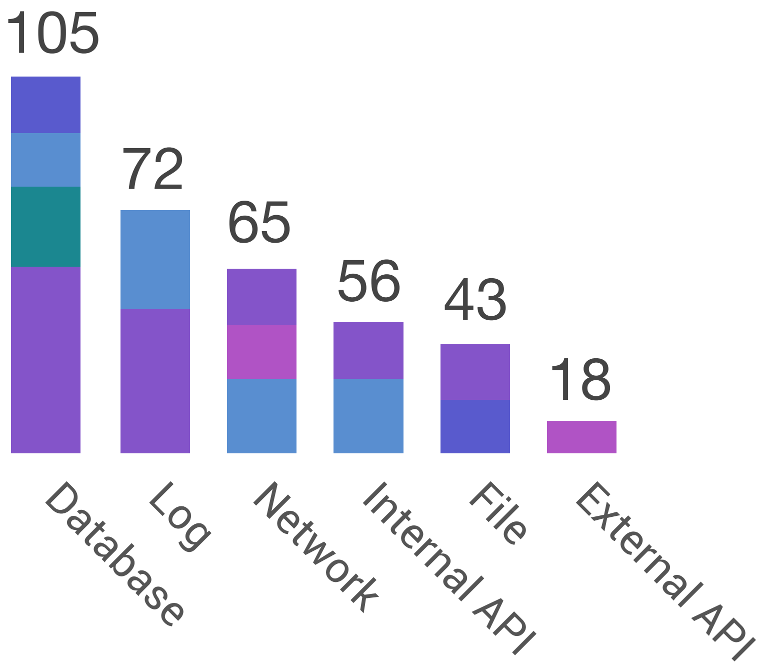
DATA CLASSIFICATION & FLOW SUMMARY



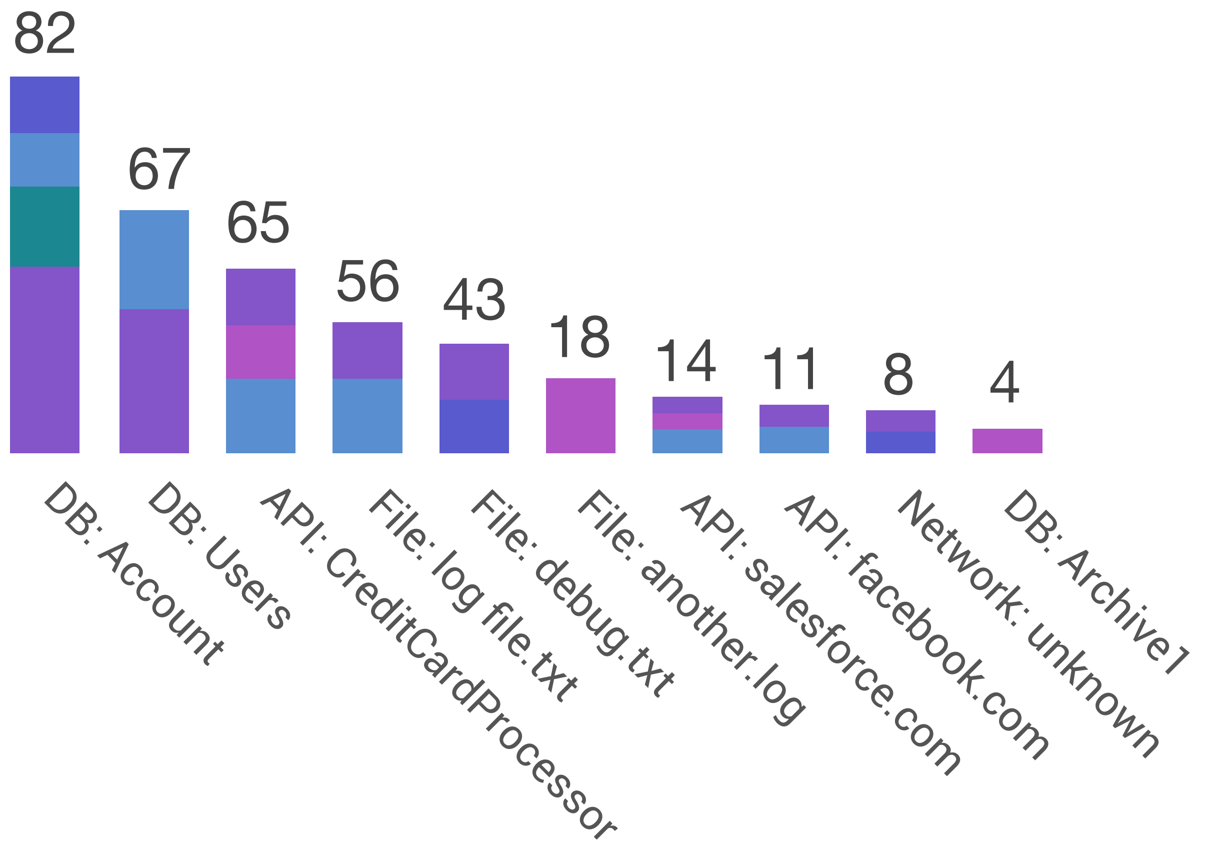
DATA SINK SUMMARY

BY NUM OF INSTANCE FLOWS

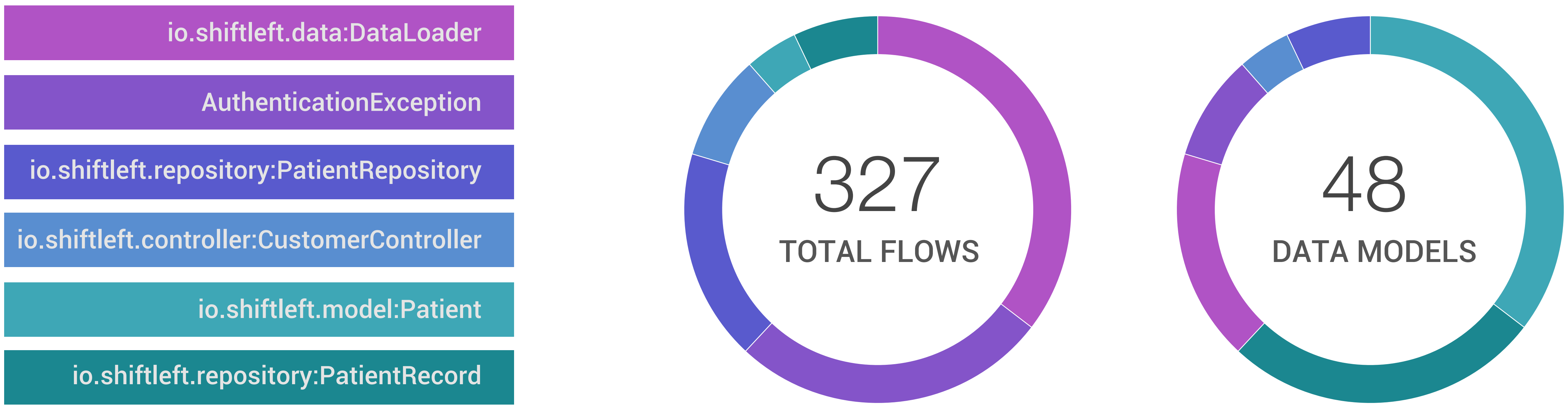
Types of sinks



All sinks



DATA MODELS: TOP 10 FLOW PARTICIPATION



DATA MODELS SUMMARY

	CLASS NAME	SENSITIVE DATA CATEGORIES
■	io.shiftleft.data:DataLoader	Pii Payment
■	org.apache.http.auth:AuthenticationException	Medical
■	io.shiftleft.repository:PatientRepository	Tokens Payment Medical
■	io.shiftleft.controller:CustomerController	Authentication Internal secrets pii
■	io.shiftleft.model:Patient	Pii Payment
■	io.shiftleft.repository:PatientRecord	Medical
	io.shiftleft.controller:CustomerController	Medical
	io.shiftleft.model:Patient	Tokens Payment
	io.shiftleft.controller:CustomerController	Authentication Internal secrets
	io.shiftleft.model:Patient	Authentication
	io.shiftleft.controller:CustomerController	Medical
	io.shiftleft.model:Patient	Tokens Payment
	io.shiftleft.controller:CustomerController	Authentication Internal secrets
	io.shiftleft.model:Patient	Authentication

SENSITIVE DATA FLOWS DETAIL

BY DATA CATEGORY

INTERNAL SECRETS

DATA DETAIL	FLOW SOURCE	TRANSFORMS	FLOW DESTINATION
TYPE **java.lang.String** VAR NAME **clientId**	io.shiftleft.controller. **CustomerController.debug()** WEB ROUTE **POST /customer/debug** LINE NUMBER **293**	1 io.shiftleft.controller. **CustomerController.debug()** - line 293 2 io.shiftleft.model. **Customer.<constructor>** - line 20	io.shiftleft.repository. **CustomerRepository.save()** LINE NUMBER **293** DATA SINK **Database**

PII

DATA DETAIL	FLOW SOURCE	TRANSFORMS	FLOW DESTINATION
TYPE **java.lang.String** VAR NAME **customerId**	io.shiftleft.controller. **CustomerController.getCustomer()** WEB ROUTE **GET /customer/<ID>** LINE NUMBER **115**	1 io.shiftleft.controller. **CustomerController.getCustomer()** - line 293	io.shiftleft.repository. **CustomerRepository.findOne()** LINE NUMBER **312** DATA SINK **Database**
TYPE **java.lang.String** VAR NAME **lastName**	io.shiftleft.controller. **CustomerController.debug()** WEB ROUTE **POST /customer/debug** LINE NUMBER **293**	1 io.shiftleft.controller. **CustomerController.debug()** - line 293 2 io.shiftleft.model. **Customer.<constructor>** - line 20	io.shiftleft.repository. **CustomerRepository.findOne()** LINE NUMBER **312** DATA SINK **Database**
TYPE **java.lang.String** VAR NAME **socialSecurityNum**	io.shiftleft.controller. **CustomerController.getCustomer()** WEB ROUTE **GET /customer/<ID>** LINE NUMBER **115**	1 io.shiftleft.controller. **CustomerController.getCustomer()** - line 293 2 io.shiftleft.model. **Customer.<constructor>** - line 20	io.shiftleft.repository. **CustomerRepository.findOne()** LINE NUMBER **312** DATA SINK **Database**

MEDICAL

DATA DETAIL	FLOW SOURCE	TRANSFORMS	FLOW DESTINATION
TYPE **io.shiftleft.controller. PatientController** VAR NAME **patient**	io.shiftleft.controller. **PatientController.getPatient()** WEB ROUTE **GET /patient/<id>** LINE NUMBER **34**	1 io.shiftleft.controller. **PatientController.getPatient()** - line 34 2 io.shiftleft.model. **Patient.toString()** - line 151	org.slf4j **Logger.info()** LINE NUMBER **78** DATA SINK **Log**