

# Java 8 in Anger

Trisha Gee

Developer & Technical Advocate, JetBrains

# Java 8 has many new features

## Java Programming Language

- Lambda Expressions, a new language feature, has been introduced in this release. They enable you to treat functionality as a method argument, or code as data. Lambda expressions let you express instances of single-method interfaces (referred to as functional interfaces) more compactly.
- Method references provide easy-to-read lambda expressions for methods that already have a name.
- Default methods enable new functionality to be added to the interfaces of libraries and ensure binary compatibility with code written for older versions of those interfaces.
- Repeating Annotations provide the ability to apply the same annotation type more than once to the same declaration or type use.
- Type Annotations provide the ability to apply an annotation anywhere a type is used, not just on a declaration. Used with a pluggable type system, this feature enables improved type checking of your code.
- Improved type inference.
- Method parameter reflection.

## Collections

- Classes in the new `java.util.stream` package provide a Stream API to support functional-style operations on streams of elements. The Stream API is integrated into the Collections API, which enables bulk operations on collections, such as sequential or parallel map-reduce transformations.
- Performance Improvement for HashMaps with Key Collisions

## Compact Profiles

### Security

- Client-side TLS 1.2 enabled by default
- New variant of `AccessController.doPrivileged` that enables code to assert a subset of its privileges, without preventing the full traversal of the stack to check for other permissions
- Stronger algorithms for password-based encryption
- SSL/TLS Server Name Indication (SNI) Extension support in JSSE Server
- Support for AEAD algorithms: The SunJCE provider is enhanced to support AES/GCM/NoPadding cipher implementation as well as GCM algorithm parameters. And the SunJSSE provider is enhanced to support AEAD mode based cipher suites. See Oracle Providers Documentation, JEP 115.
- KeyStore enhancements, including the new Domain KeyStore type `java.security.DomainLoadStoreParameter`, and the new command option `-importpassword` for the keytool utility
- SHA-224 Message Digests
- Enhanced Support for NSA Suite B Cryptography
- Better Support for High Entropy Random Number Generation
- New `RevocationChecker` and `RevocationCheckerImpl` classes for configuring revocation checking of X.509 certificates
- New 64-bit PKCS11 for Windows
- New cache types in Kerberos 5 Replay Caching
- Support for Kerberos 5 Protocol Transition and Constrained Delegation
- Kerberos 5 weak encryption types disabled by default
- Inbound SASL for the GSS-API/Kerberos 5 mechanism
- SASL service for multiple host names
- JNI bridge to native JGSS on Mac OS X
- Support for stronger strength ephemeral DH keys in the SunJSSE provider
- Support for server-side cipher suites preference customization in JSSE

## JavaFX

- The new Modena theme has been implemented in this release. For more information, see the blog at [fxexperience.com](http://fxexperience.com).
- The new `SwingNode` class enables developers to embed Swing content into JavaFX applications. See the [SwingNode](#), [javacod](#) and [Embedding Swing Content in JavaFX Applications](#).
- The new UI Controls include the [DatePicker](#) and the [WebView](#) controls.
- The `javafx.graphics` package provides the public classes for the JavaFX Printing API. See the [javacod](#) for more information.
- The 3D Graphics features now include 3D shapes, camera, lights, subscene, material, picking, and antialiasing. The new `Shape3D` (`Box`, `Cylinder`, `MeshView`, and `Sphere` subclasses), `SubScene`, `Material`, `PickResult`, `LightBase` (`AmbientLight` and `PointLight` subclasses), and `SceneAntialiasing` API classes have been added to the JavaFX 3D Graphics library. The `Camera` API class has also been updated in this release. See the corresponding class [javacod](#) for more information.
- The `WebView` class has new features and improvements. Review [Supported Features of HTML5](#) for more information about additional HTML5 features including Web Sockets, Web Workers, and Web Fonts.
- Enhanced text support including bi-directional text and complex text scripts such as Thai and Hindi in controls, and multi-line, multi-style text in text nodes.
- Support for Hi-DPI displays has been added in this release.
- The "CSS Styable" classes became public API. See the [JavaFX.css](#) [javacod](#) for more information.
- The new `SchedulerService` class allows to automatically restart the service.
- JavaFX is now available for ARM platforms. JDK for ARM includes the base, graphics and controls components of JavaFX.

## Tools

- The `java` command is provided to invoke the Nashorn engine.
- The `javac` command launches JavaFX applications.
- The `java` man page has been rewritten.
- The `jdeps` command-line tool is provided for analyzing class files.
- Java Management Extensions (JMX) provide remote access to diagnostic commands.
- The `javacodes` tool has an option for requesting a signed time stamp from a Time Stamping Authority (TSA).
- [javac tool](#)
  - The `-parameters` option of the `javac` command can be used to store formal parameter names and enable the Reflection API to retrieve formal parameter names.
  - The type rules for equality operators in the Java Language Specification (JLS) Section 15.21 are now correctly enforced by the `javac` command.
  - The `javac` tool now has support for checking the content of `javacod` comments for issues that could lead to various problems, such as invalid HTML or accessibility issues, in the files that are generated when `javacod` is run. The feature is enabled by the new `-xdoclint` option. For more details, see the output from running `*javac -X`. This feature is also available in the `javacod` tool, and is enabled there by default.
  - The `javac` tool now provides the ability to generate native headers, as needed. This removes the need to run the `javah` tool as a separate step in the build pipeline. The feature is enabled in `javac` by using the new `-h` option, which is used to specify a directory in which the header files should be written. Header files will be generated for any class which has either native methods, or constant fields annotated with a new annotation of type `java.lang.annotation.Native`.
- [javacod tool](#)
  - The `javacod` tool supports the new `DocTree` API that enables you to traverse `javacod` comments as abstract syntax trees.
  - The `javacod` tool supports the new `javacod` Access API that enables you to invoke the `javacod` tool directly from a Java application, without executing a new process. See the [javacod what's new](#) page for more information.
  - The `javacod` tool now has support for checking the content of `javacod` comments for issues that could lead to various problems, such as invalid HTML or accessibility issues, in the files that are generated when `javacod` is run. The feature is enabled by default, and can also be controlled by the new `-xdoclint` option. For more details, see the output from running `*javacod -X`. This feature is also available in the `javac` tool, although it is not enabled by default.

## Internationalization

- Unicode Enhancements, including support for Unicode 6.2.0
- Adoption of Unicode CLDR Data and the `java.locale.providers` System Property
- New Calendar and Locale APIs
- Ability to Install a Custom Resource Bundle as an Extension

## Deployment

- For sandbox applets and Java Web Start applications, `URLPermission` is now used to allow connections back to the server from which they were started. `SocketPermission` is no longer granted.
- The `Permissions` attribute is required in the JAR file manifest of the main JAR file at all security levels.
- A new set of packages that provide a comprehensive date-time model.

## Date-Time Package

### Scripting

### Pack200

- Pack200 Support for Constant Pool Entries and New Bytecodes Introduced by JSR 292
- JDK8 support for class files changes specified by JSR-292, JSR-308 and JSR-335

## IO and NIO

- New `SelectorProvider` implementation for Solaris based on the Solaris event port mechanism. To use, run with the system property `java.nio.channels.spi.Selector.set` to the value `sun.nio.ch.EventPortSelectorProvider`.
- Decrease in the size of the `<JDK_HOME>/jre/lib/charsets.jar` file
- Performance improvement for the `java.lang.String(byte[], *)` constructor and the `java.lang.String.getBytes()` method.

## java.lang and java.util Packages

- Parallel Array Sorting
- Standard Encoding and Decoding Base64
- Unsigned Arithmetic Support

## JDBC

- The JDBC-ODBC Bridge has been removed.
- JDBC 4.2 introduces new features.

## Java DB

- JDK 8 includes Java DB 10.10.

## Networking

- The class `java.net.URLPermission` has been added.
- In the class `java.net.HttpURLConnection`, if a security manager is installed, calls that request to open a connection require permission.

## Concurrency

- Classes and interfaces have been added to the `java.util.concurrent` package.
- Methods have been added to the `java.util.concurrent.ConcurrentHashMap` class to support aggregate operations based on the newly added streams facility and lambda expressions.
- Classes have been added to the `java.util.concurrent.atomic` package to support scalable updatable variables.
- Methods have been added to the `java.util.concurrent.ForkJoinPool` class to support a common pool.
- The `java.util.concurrent.locks.StampedLock` class has been added to provide a capability-based lock with three modes for controlling read/write access.

## Java XML - JAXP

### HSR201

- Hardware intrinsics were added to use Advanced Encryption Standard (AES). The `UseAES` and `UseAESIntrinsics` flags are available to enable the hardware-based AES intrinsics for Intel hardware. The hardware must be 2010 or newer Westmere hardware. For example, to enable hardware AES, use the following flags:

-XX:+UseAES -XX:+UseAESIntrinsics

To disable hardware AES use the following flags:

-XX:-UseAES -XX:-UseAESIntrinsics

Removal of PermGen.

Default Methods in the Java Programming Language are supported by the byte code instructions for method invocation.

## Java Mission Control 5.3 Release Notes

- JDK 8 includes Java Mission Control 5.3.

# Lambdas

(...and Streams)

# Real World Application



TWITTER  
FIREHOSE

CONNECT  
&  
EXTRACT

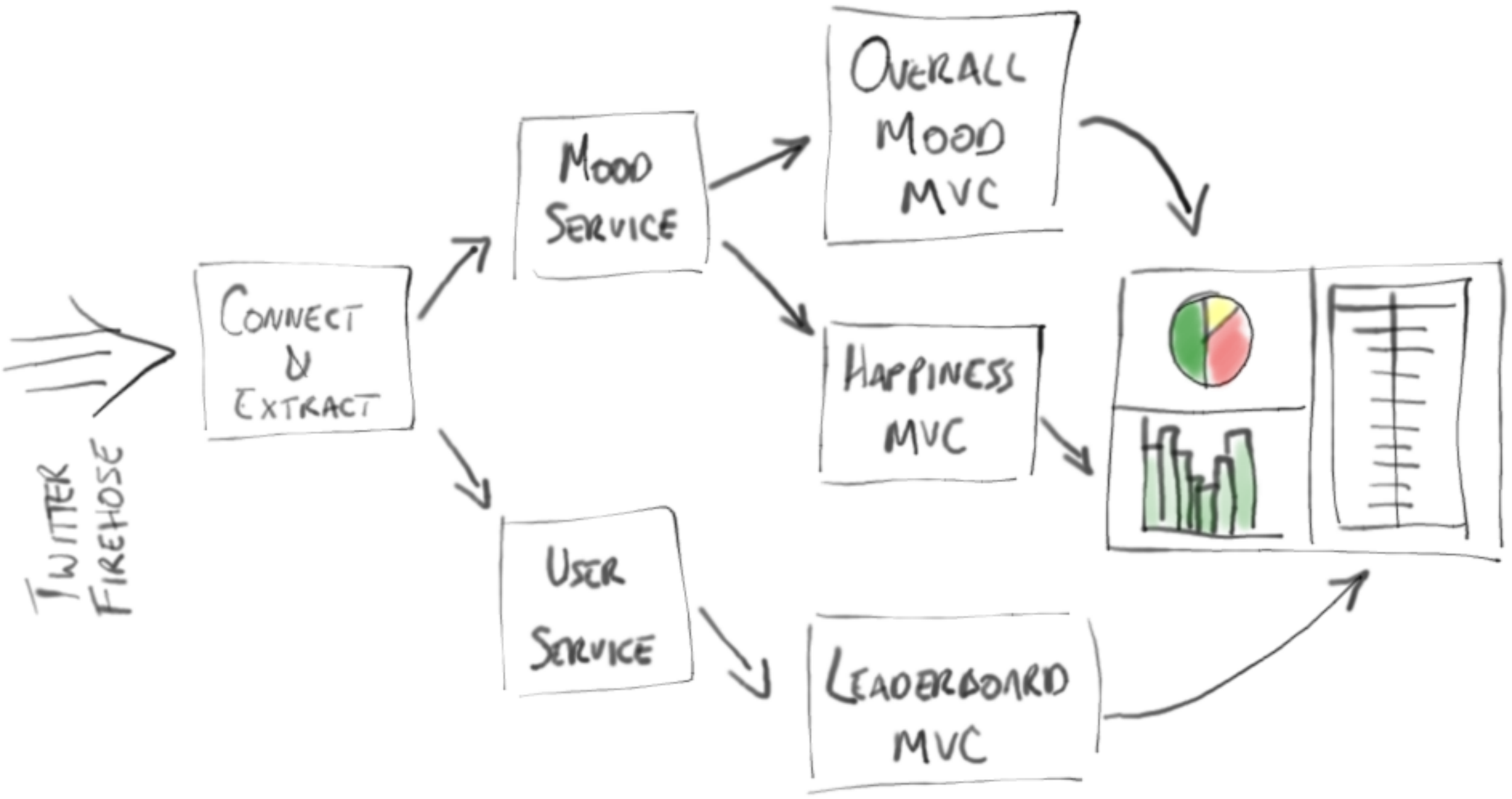
MOOD  
SERVICE

OVERALL  
MOOD  
MVC

HAPPINESS  
MVC

USER  
SERVICE

LEADERBOARD  
MVC



# Play Along

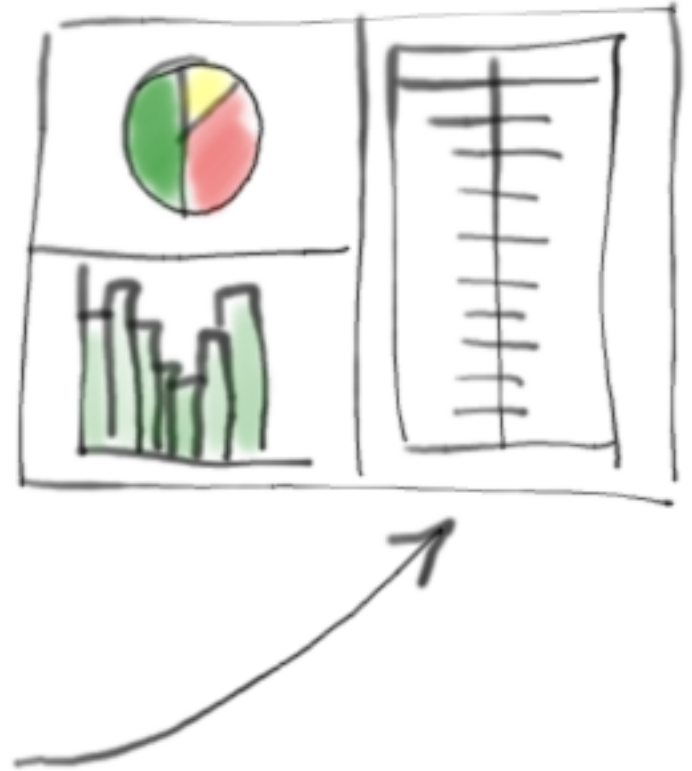
<https://github.com/trishagee/sense>



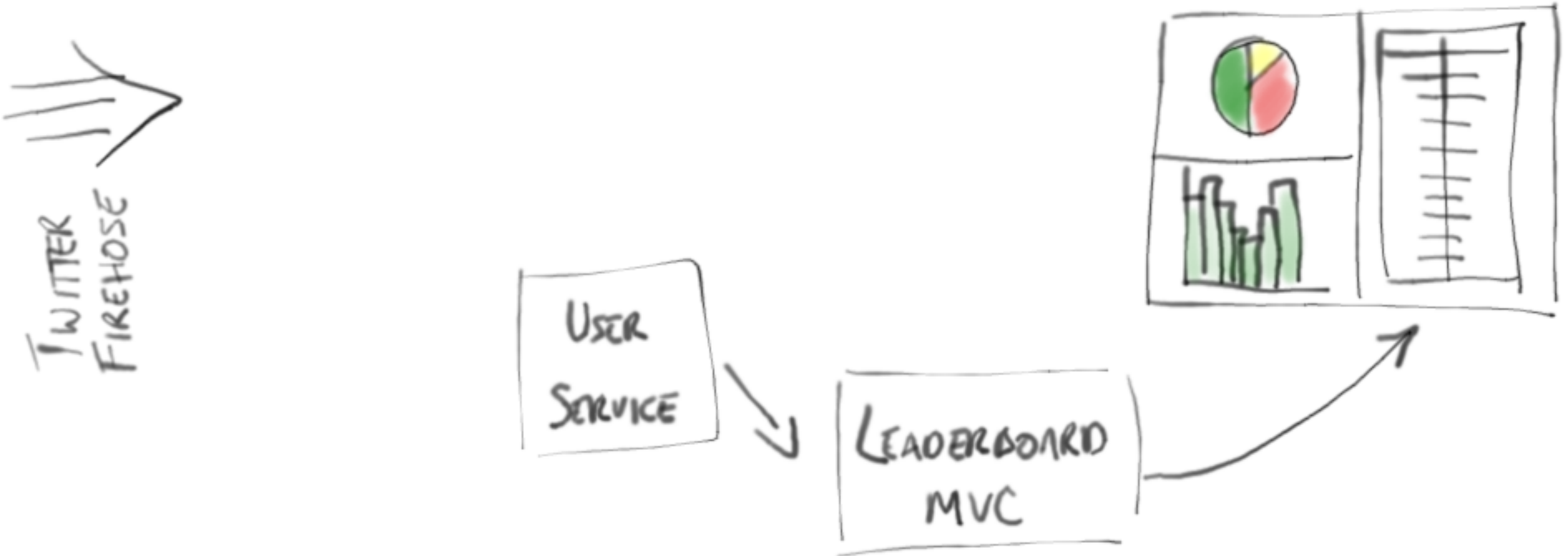
# Dashboard



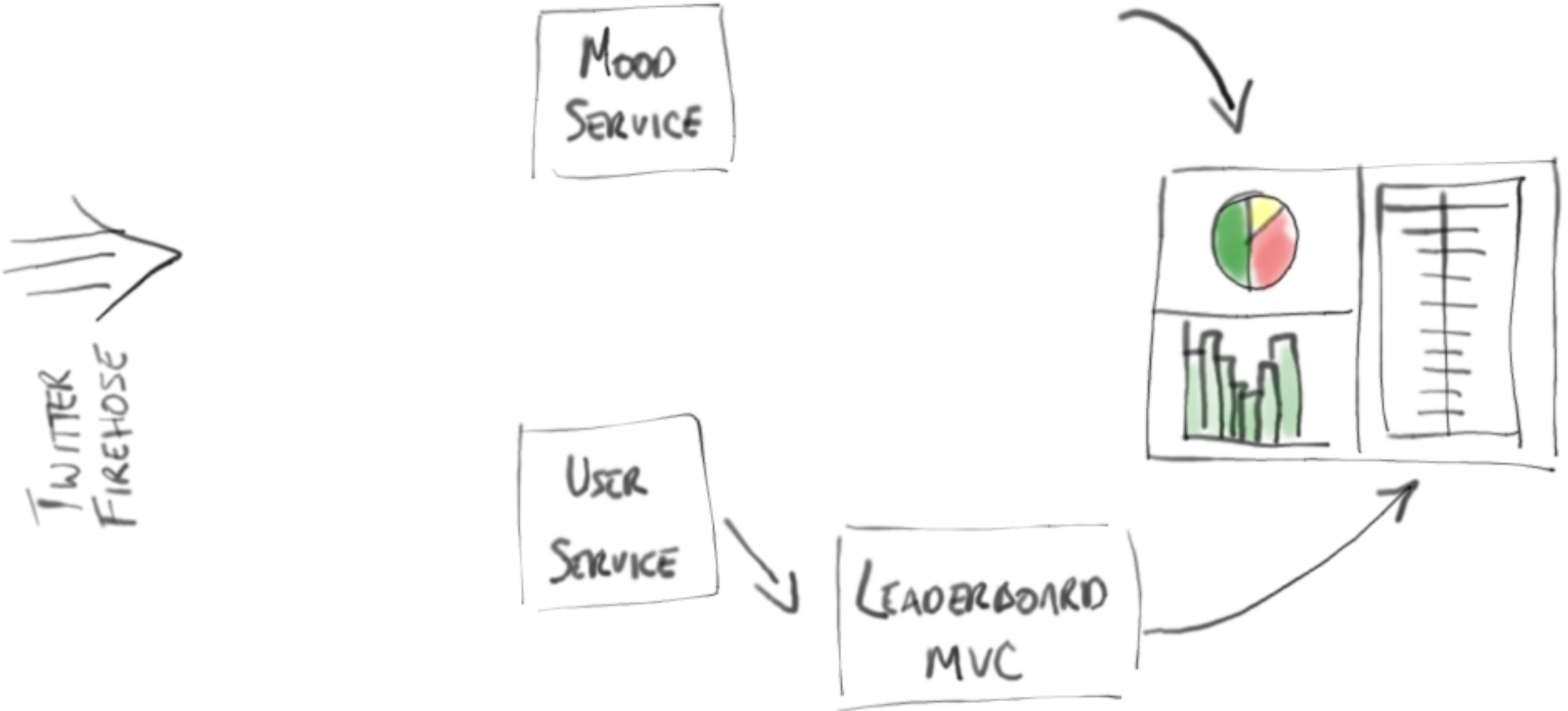
# Twitter Leaderboard



# Twitter Leaderboard

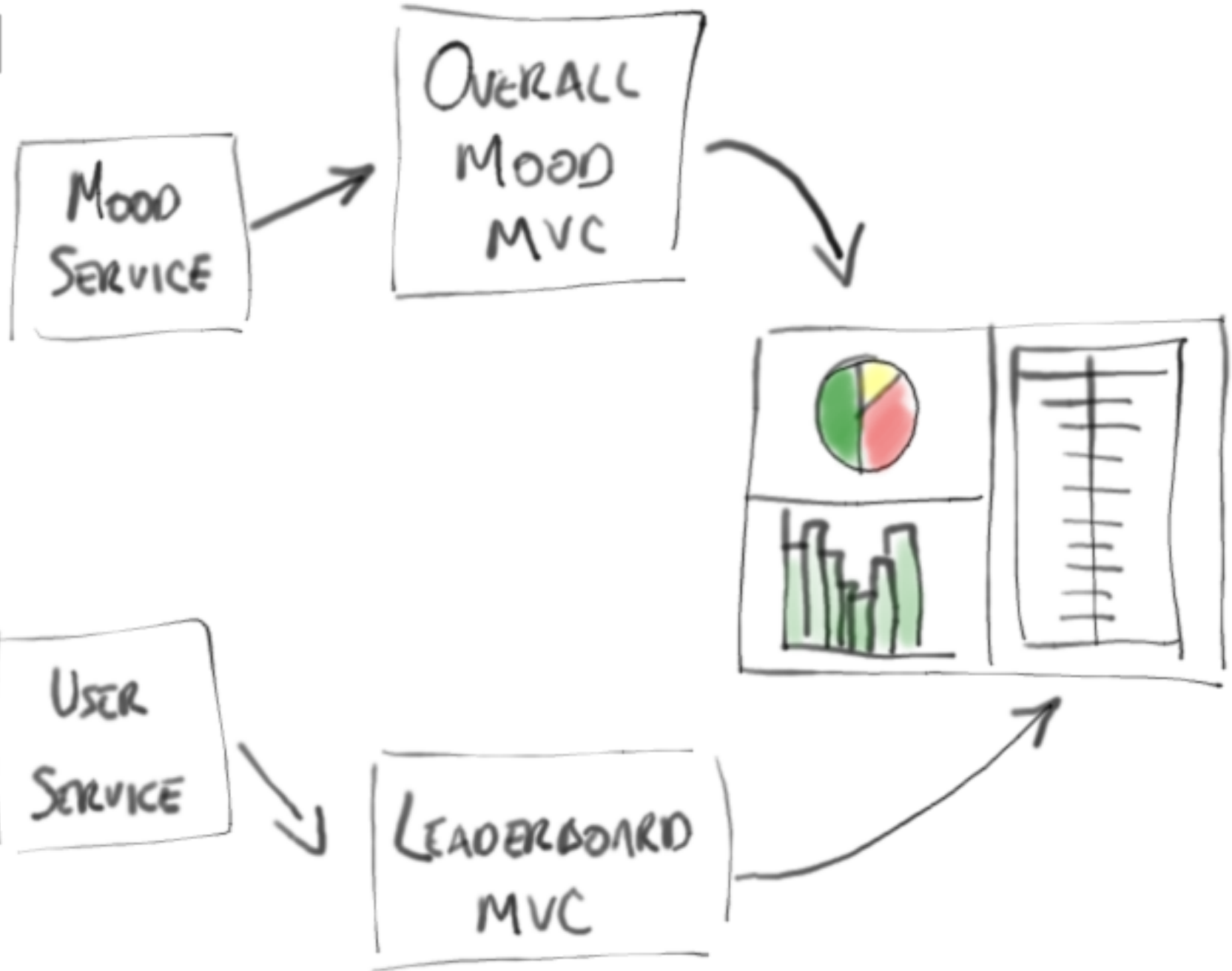


# Overall Mood



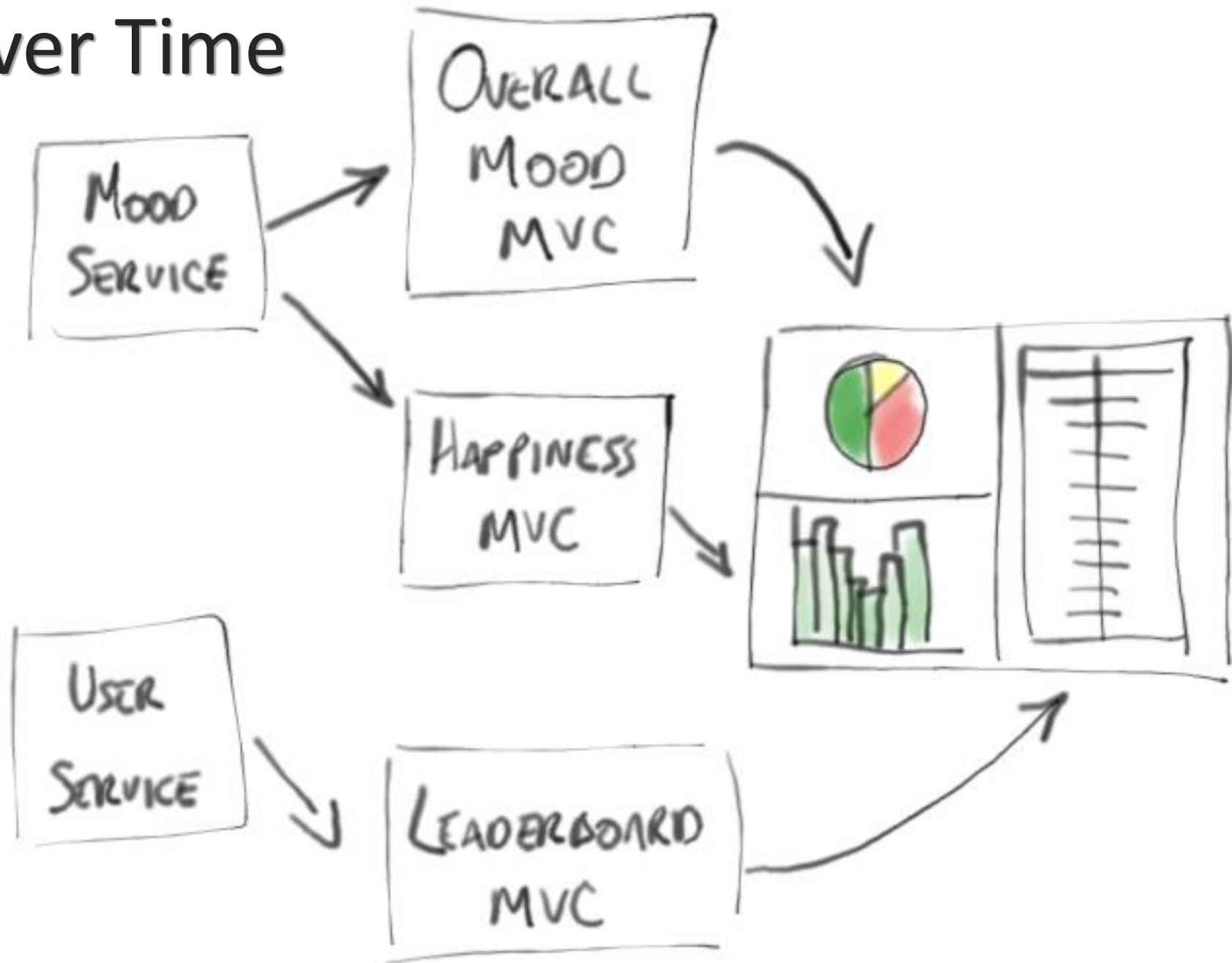
# Overall Mood

TWITTER  
FIREHOSE

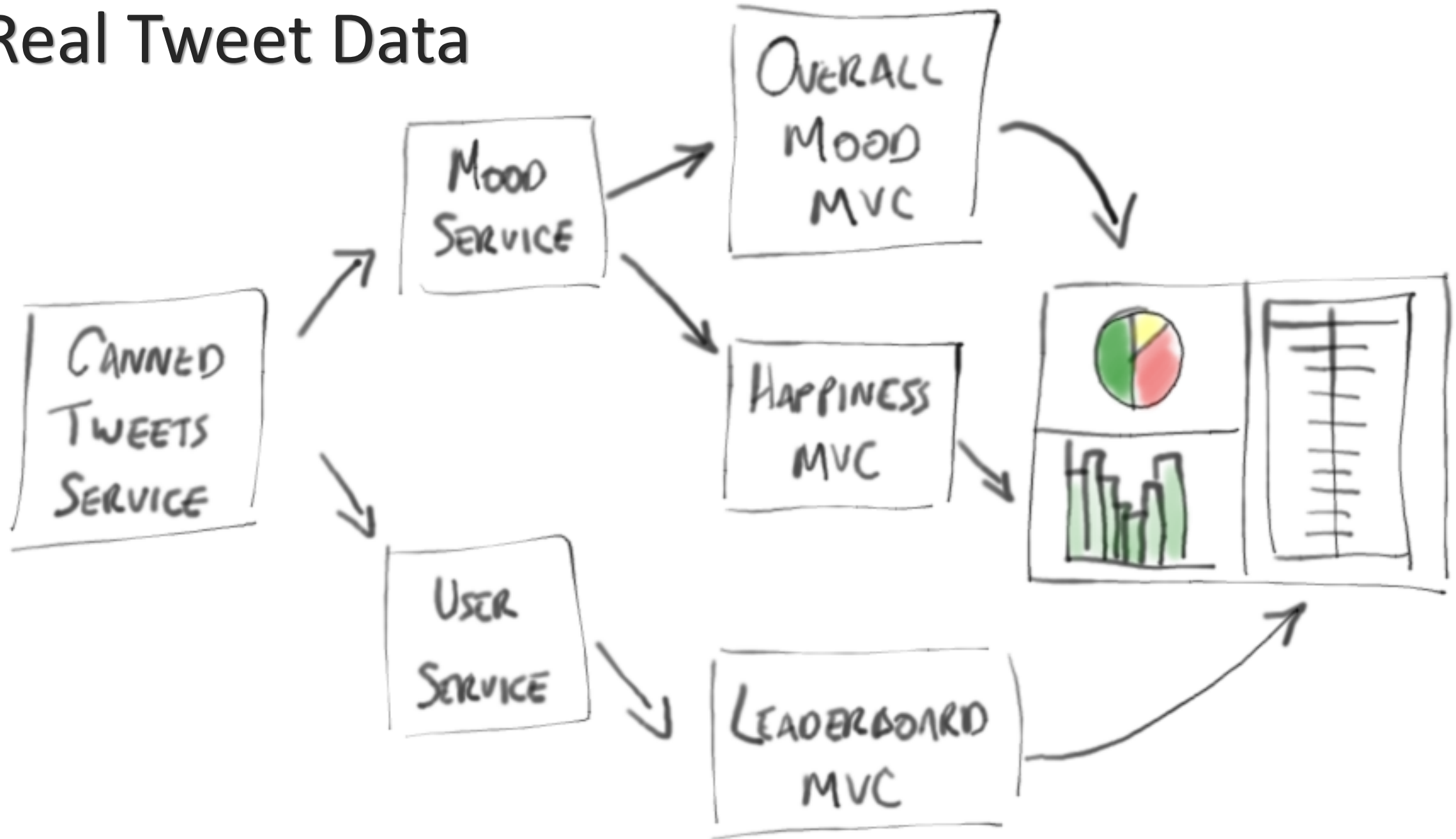


# Happiness Over Time

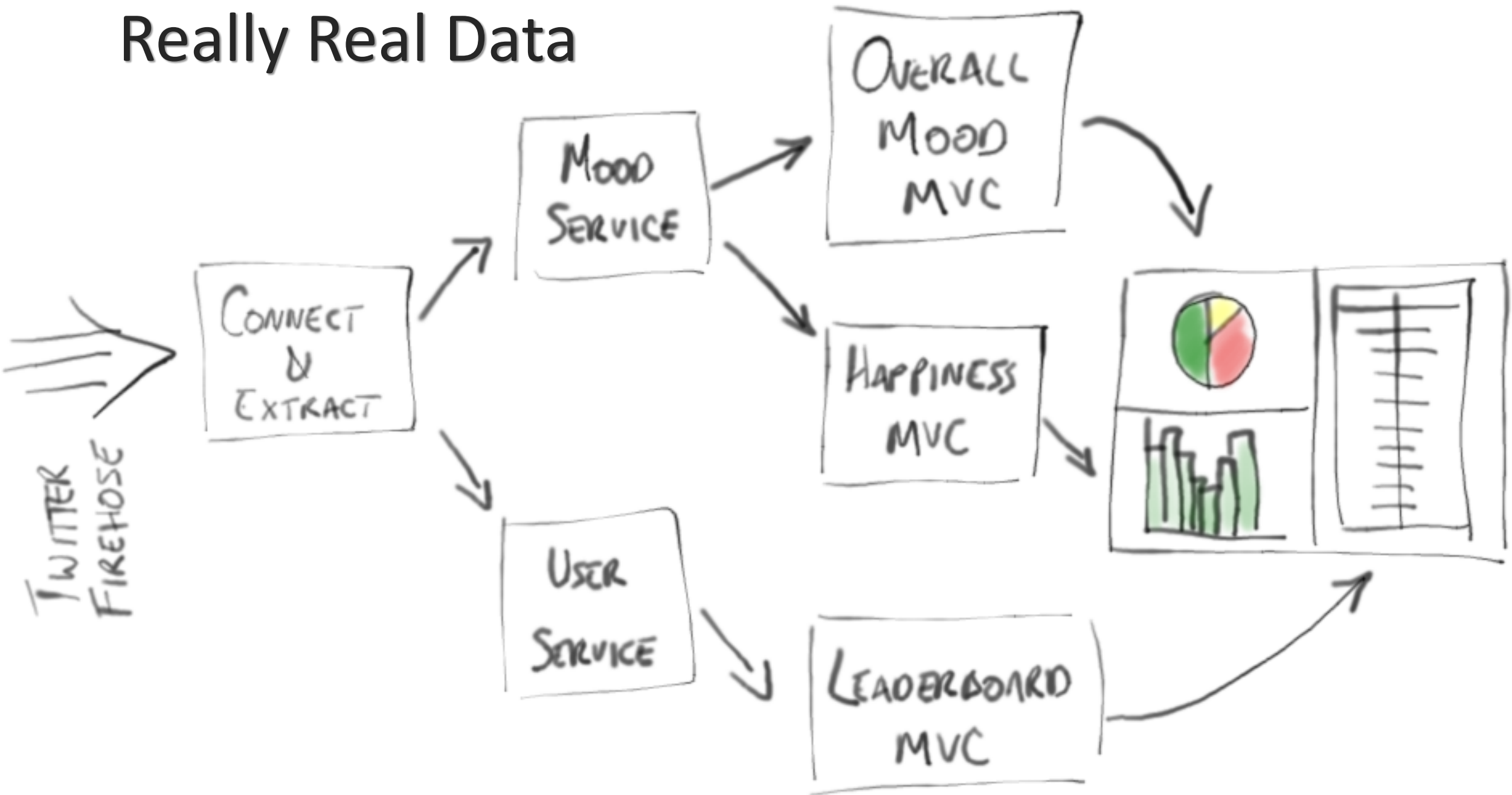
TWITTER  
FIREHOSE



# Real Tweet Data



# Really Real Data





What have we learnt?

# What have we learnt?

- How to use Streams to manipulate data
- Where Lambdas might simplify your code
- Basics of building a JavaFX UI

# Resources

<http://bit.ly/Java8InAnger>

# Questions?

@trisha\_gee

<http://bit.ly/Java8InAnger>