



Simba  
t e c h n o l o g i e s

# ENSC 251: How Simba does Software

**Sylvia Lee & Maria Kaardal**

Simba Technologies

Dec 2, 2015



Simba  
t e c h n o l o g i e s

is the...



Choose your metaphor.

# Identify the Use Case

## Case

- What problem does your software need to solve?



Get me my data from \_\_\_\_\_



THE BEST  
DECISION  
POSSIBLE™

# Identify the Customer

# Meet Dale, Hank, Cynthia, and Andrea



Dale, Developer



Hank, Dev Manager



Cynthia, CTO/CIO



Andrea, Analyst



# What Does Andrea need?






# Know thy Customer


- Database/Data source vendor?
- BI/App vendor?
- End user?
- Engineer?

## Connectors


Get connected quickly with your data for comprehensive business intelligence without the need for development. With Simba's easy, scalable and supported solution, why build it yourself.

 [Learn more](#)


ODBC/JDBC [Buy NOW](#) [FREE Trial](#)

 [Learn more](#)


ODBC/JDBC [Buy NOW](#) [FREE Trial](#)

 [Learn more](#)

ODBC/JDBC [Buy NOW](#) [FREE Trial](#)


 [Learn more](#)

ODBC/JDBC [Buy NOW](#) [FREE Trial](#)

 [Learn more](#)


ODBC [Buy NOW](#) [FREE Trial](#)

JDBC [Contact Us](#) +1 (804) 633-0008

 [Learn more](#)


ODBC [Buy NOW](#) [FREE Trial](#)

JDBC [Contact Us](#) +1 (804) 633-0008

 [Learn more](#)


ODBC [Buy NOW](#) [FREE Trial](#)

JDBC [Contact Us](#) +1 (804) 633-0008


 [Learn more](#)

ODBC [Buy NOW](#) [FREE Trial](#)

JDBC [Contact Us](#) +1 (804) 633-0008

 [Learn more](#)

ODBC/JDBC [Buy NOW](#) [FREE Trial](#)

 [Learn more](#)

ODBC/JDBC [Buy NOW](#) [FREE Trial](#)



### SimbaEngine SDK

Relational Data Connectivity - ODBC, JDBC, ADO.NET and OLE DB

The premier SQL connector solution in the market today SimbaEngine SDK provides a complete, open framework for quick implementation of a custom data driver for your SQL, legacy, non-relational or proprietary data store. Provides a quick and easy way to enable ODBC, JDBC, ADO.NET or OLE DB data access to your SQL-capable, proprietary or non-relational data store.

[View Product](#)

[FREE Trial](#)



### SimbaProvider SDK

OLAP / Multi-dimensional Data Connectivity / MDX

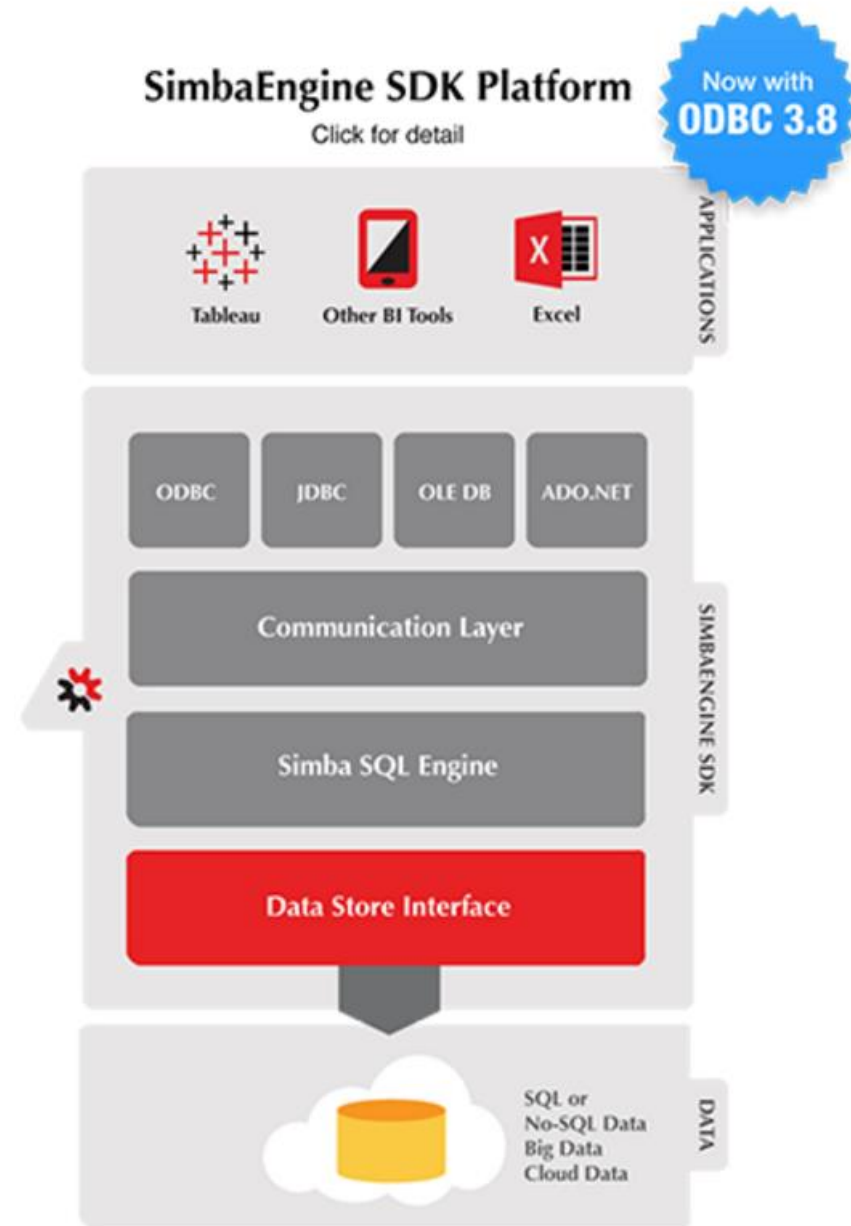
Industry's Leading MDX and OLAP SDK for ODBO and XMLA Data Access

Connect to multi-dimensional and star-schema relational data sources. Integrate your data source with the newest generation of analytical applications and development environments, such as SAP BusinessObjects, IBM Cognos, Microsoft Excel, ADOMD, ADOMD.NET and many others.

[View Product](#)

[FREE Trial](#)

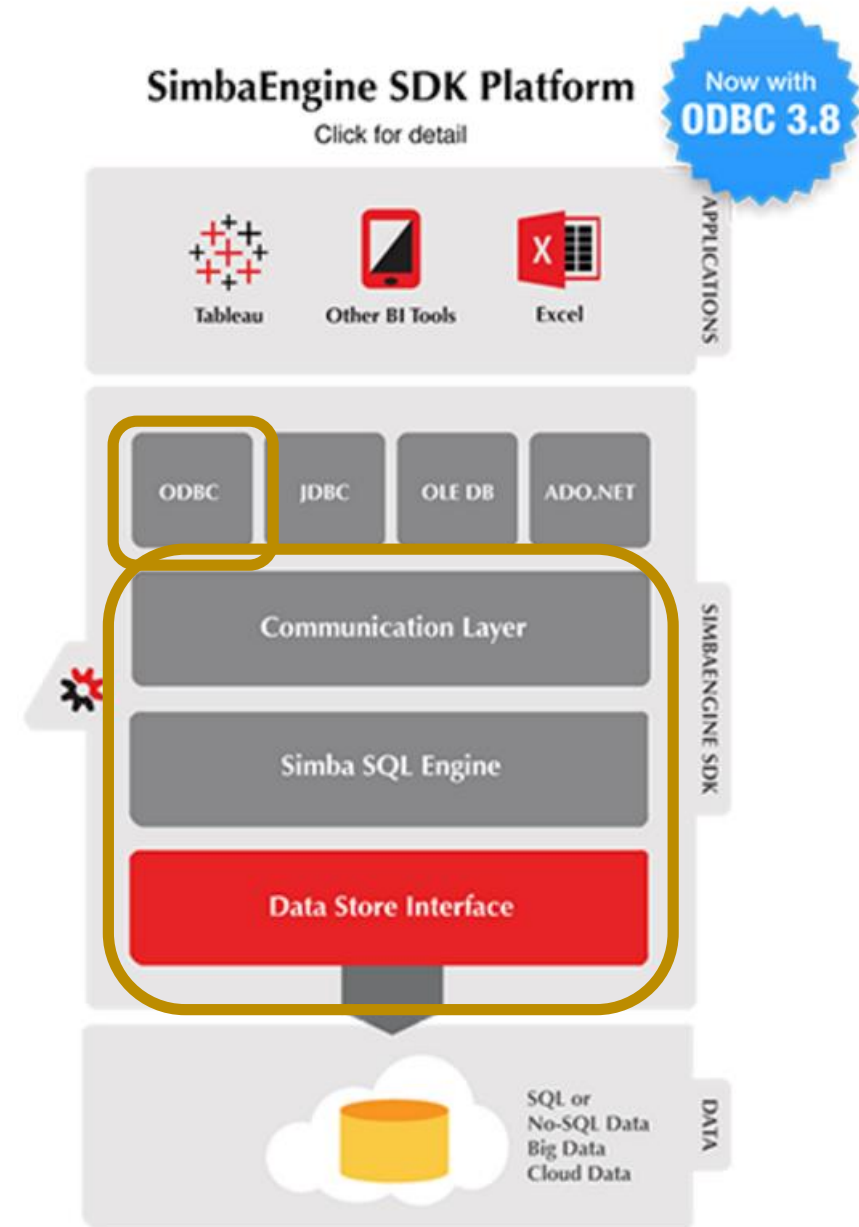
# Simba Engine SDK



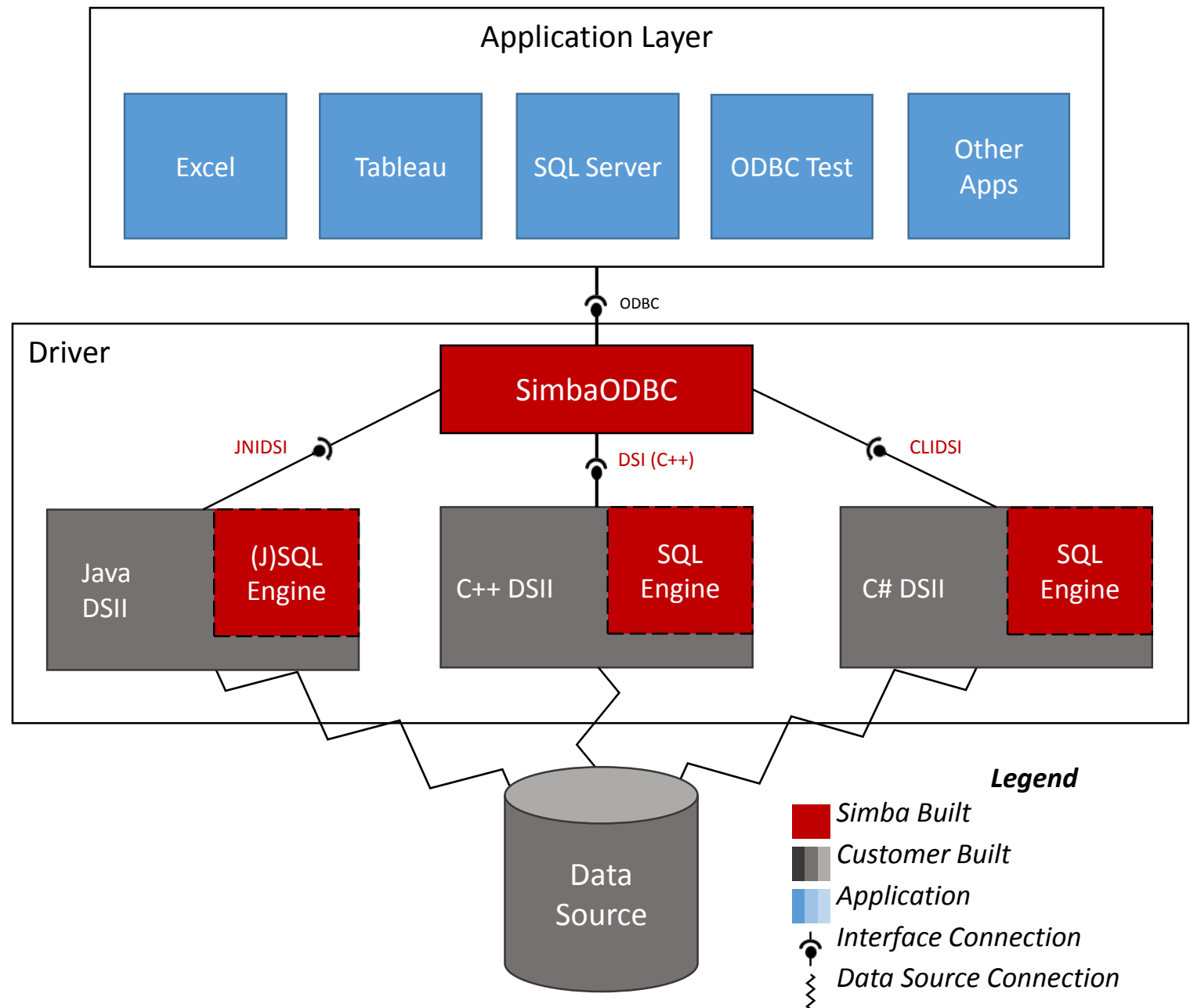
# Let's Recap

- Huge variety of data sources
  - Customers want to connect to data source X.
  - Customers can be SDK customers or Connector customers.
- 
- How can we design our SDKs so that they are flexible enough to connect to *any* data source?

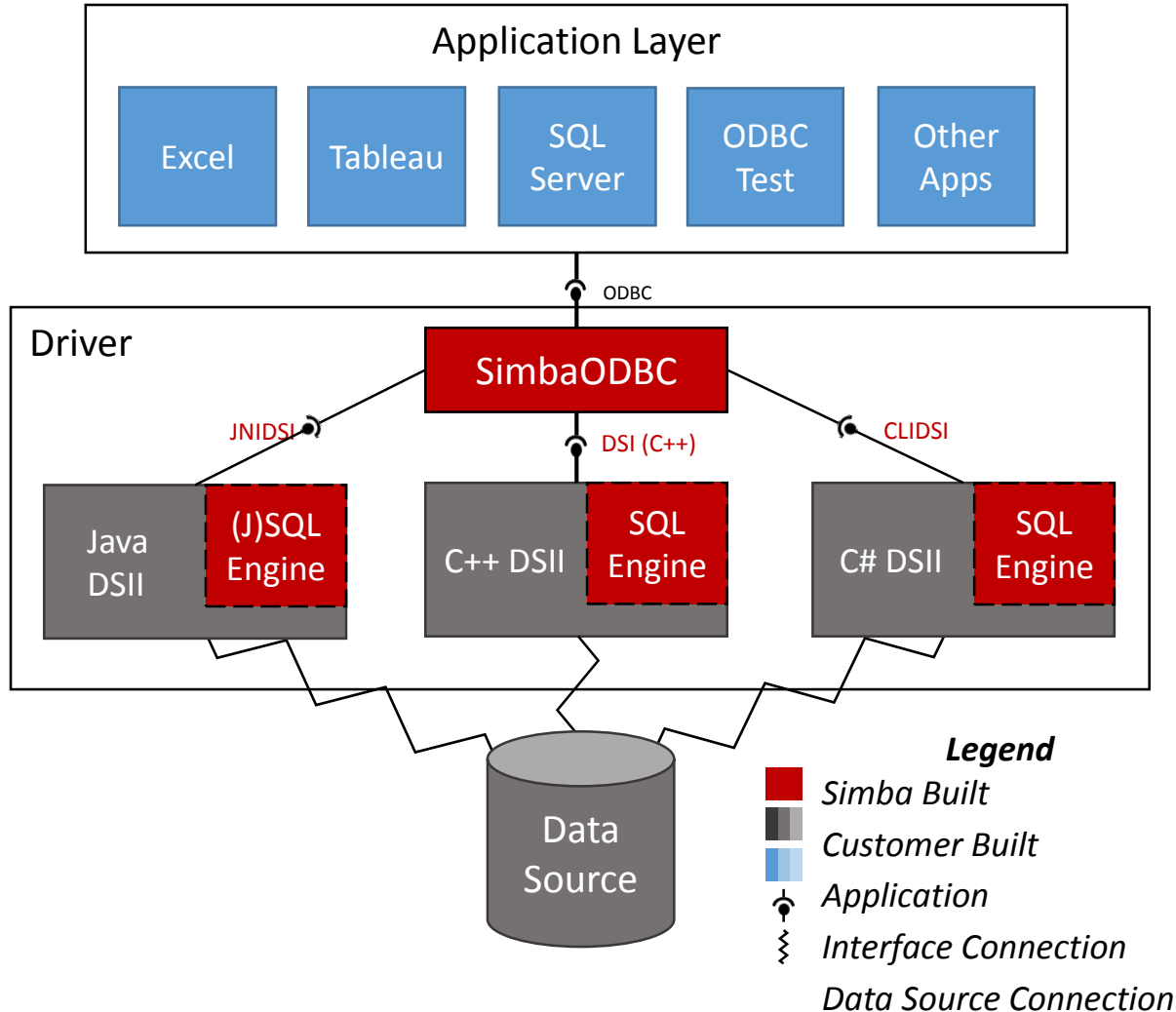
# ODBC Stack in Closer Detail



# ODBC Stack in Closer Detail



# ODBC Stack in Closer Detail

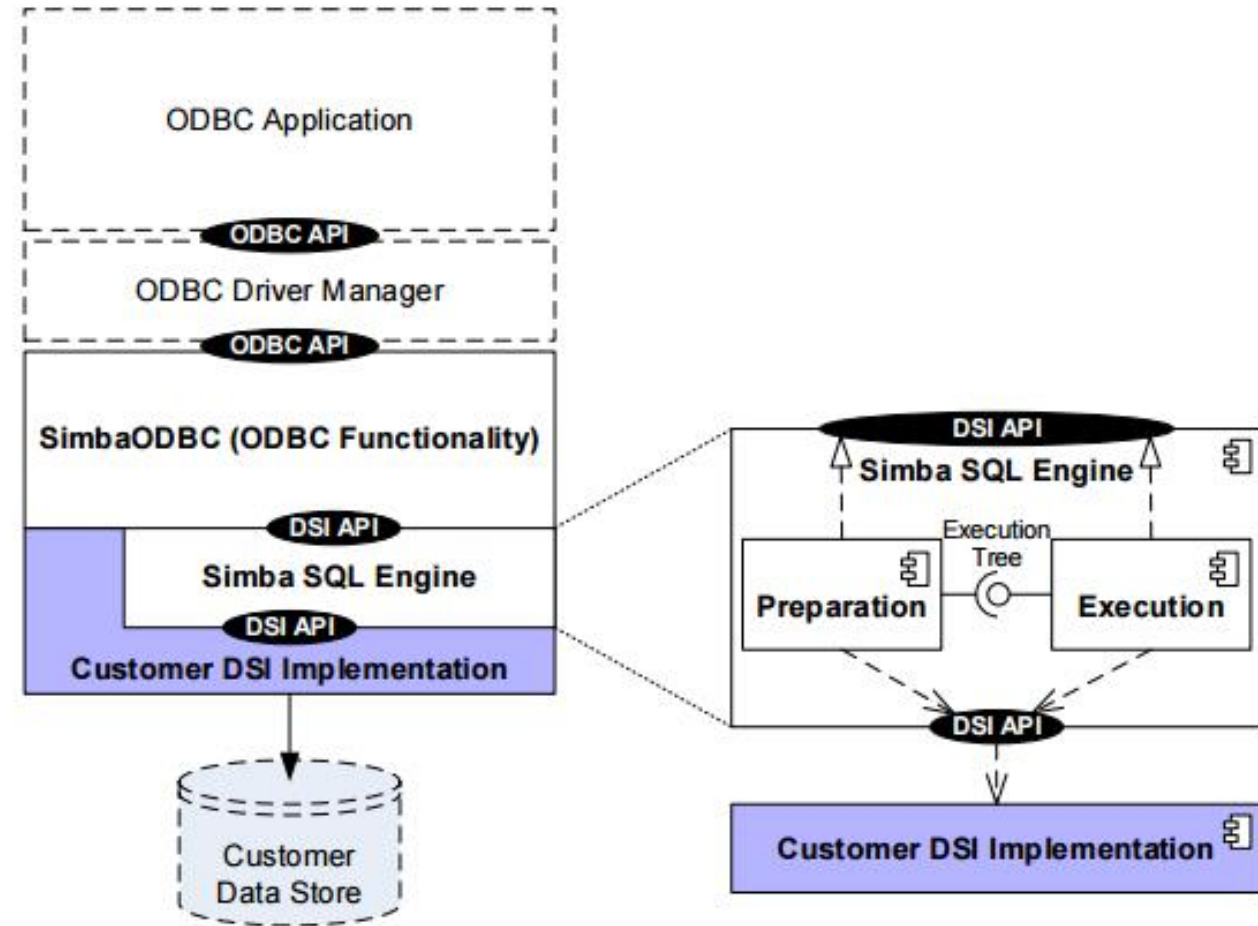


- SimbaODBC module implements ODBC interface
- SimbaODBC module exposes DSI interface
- DSII layer implements DSI interface
- Three languages – Java, C++, C#

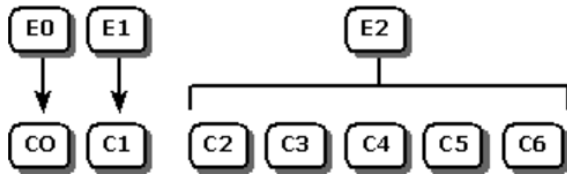


# Data Store Interface (DSI)

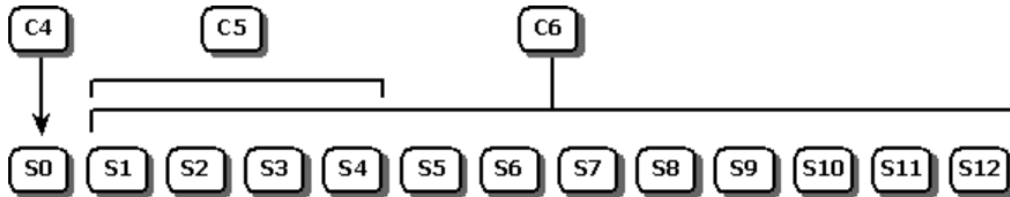
- The DSI Layer interfaces between the ODBC Layer and the Data Store Interface Implementation (DSII) built by the customer.



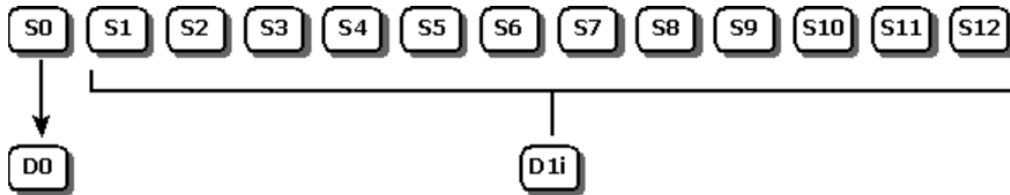
# Why Simba Engine SDK?



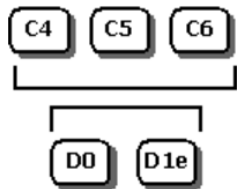
The connection and statement states overlap as follows:



The statement and descriptor states overlap as follows:



The connection and descriptor states overlap as follows:



## ▼ ODBC API Reference

[SQLAllocConnect Function](#)

[SQLAllocEnv Function](#)

[SQLAllocHandle Function](#)

[SQLAllocStmt Function](#)

[SQLBindCol Function](#)

[SQLBindParameter Function](#)

[SQLBrowseConnect Function](#)

[SQLBulkOperations Function](#)

[SQLCancel Function](#)

[SQLCancelHandle Function](#)

[SQLCloseCursor Function](#)

[SQLColAttribute Function](#)

[SQLColAttributes Function](#)

[SQLColumnPrivileges Function](#)

[SQLColumns Function](#)

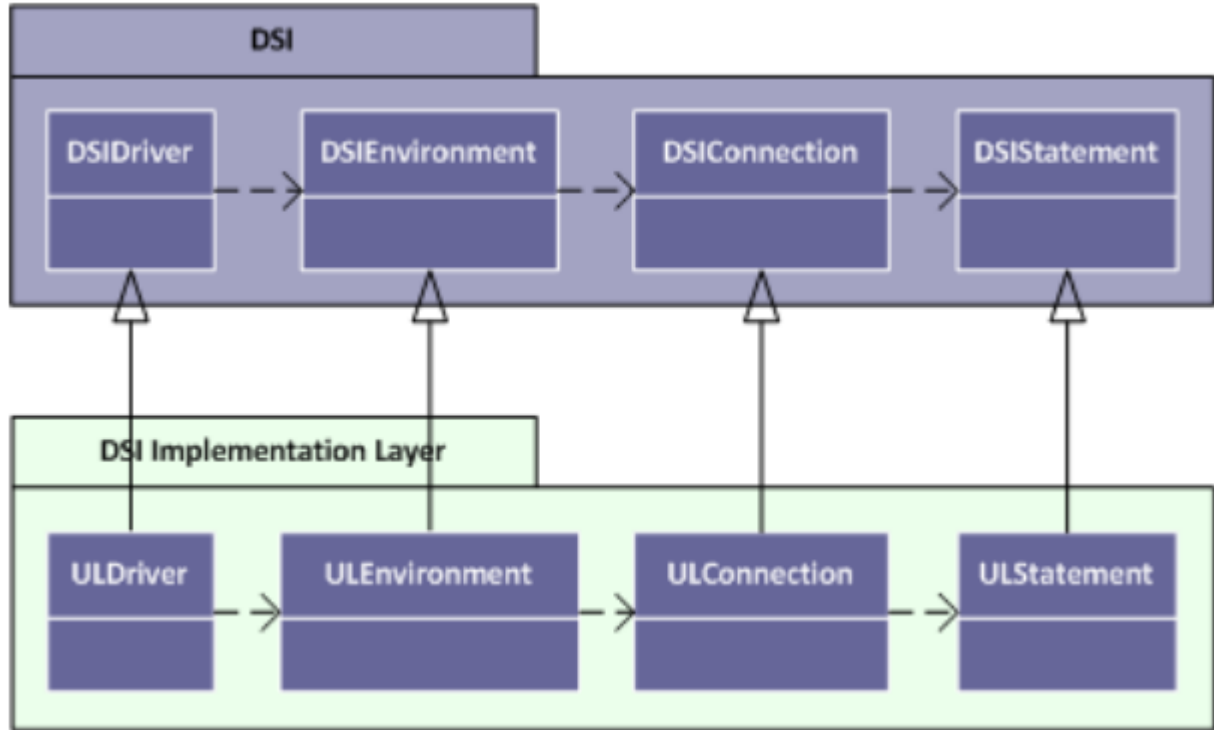
# An Object-Oriented Approach

Drivers need to be able to execute queries:

- *SELECT \* FROM T1;*
- *INSERT INTO T1 VALUES(?)*
- *CREATE TABLE T1(c1 CHAR(1));*

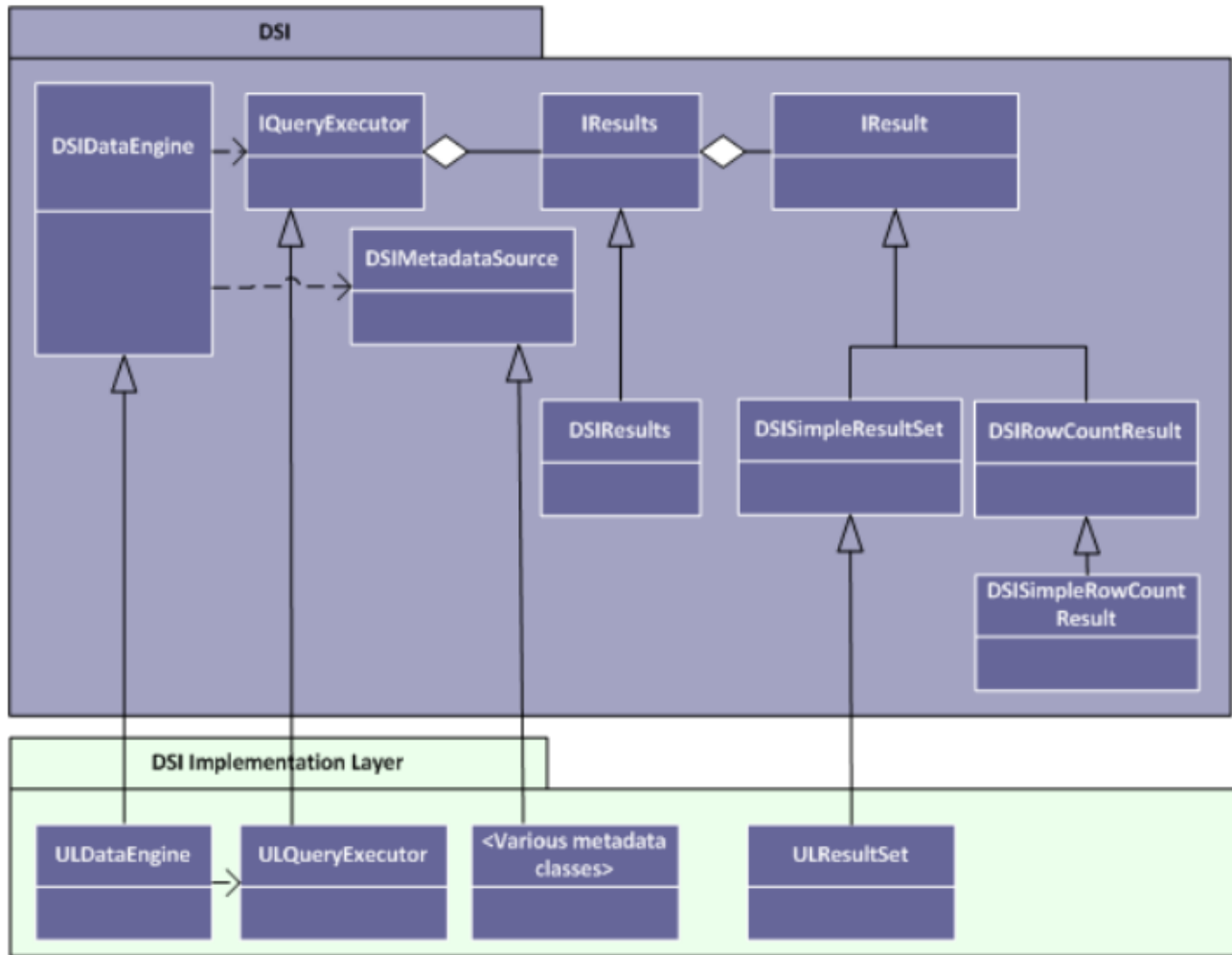
- Driver
- Environment
- Connection
- Statement
- DataEngine
- QueryExecutor
- ResultSet
- RowCount

# Core DSI Classes



- The DSII derives each of the core DSI classes – Driver, Environment, Connection, Statement
- Driver creates Environments
- Environments create Connections
- Connections create Statements

# DataEngine DSI Classes



- Statements create DataEngines
- DataEngines create QueryExecutors.
- QueryExecutors create Results.
- Results are collections of Result objects.

# Ownership, Ownership, Ownership!

- It ***MUST*** be clear what owns what objects so that you don't end up either:
  - Leaking memory
  - Deleting what you don't own
- Connections create Statements. The question is, what owns the created Statement object?

```
virtual Simba::DSI::IStatement* CreateStatement ( ) [pure virtual]
```

Creates and returns a new **IStatement** instance.

**Returns:**

New **IStatement** instance. (OWN)

Implements **IConnection**.

# NOT OWN Example

- Another example from the DSIConnection class:

**Simba::DSI::IEnvironment\* GetParentEnvironment ( ) [virtual]**

Gets the parent environment for this connection.

**Returns:**

The parent environment instance. (NOT OWN)

Implements **IConnection**.

# DSIConnection Class Reference

This class exposes an interface to interact with the customer DSII for connection details. [More...](#)

Inherits [Simba::DSI::IConnection](#).

[List of all members](#).

## Public Member Functions

virtual void	<b>BeginTransaction</b> () <i>Begins a transaction.</i>
virtual void	<b>ClearCancel</b> () <i>Clear any previous cancel notification requested by the application.</i>
virtual void	<b>ClonePropertyMap</b> ( <a href="#">DSIConnPropertyKeyValueMap</a> &io_propertyMap) <i>Retrieves a clone of the property map for the connection.</i>
virtual void	<b>Commit</b> () <i>This method commits all transactions within this connection.</i>
virtual void	<b>Connect</b> (const <a href="#">DSIConnSettingRequestMap</a> &in_connectionSettings)=0 <i>Attempts to connect to the data source, using connection settings generated by a call to <a href="#">UpdateConnectionSettings()</a>.</i>
virtual <a href="#">Simba::DSI::IStatement</a> *	<b>CreateStatement</b> ()=0 <i>Creates and returns a new <a href="#">IStatement</a> instance.</i>
virtual void	<b>Disconnect</b> ()=0 <i>Disconnect the connection.</i>
virtual const <a href="#">ConnectionSettingInfoMap</a> &	<b>GetConnectionSettingInfo</b> () <i>Gets the <a href="#">ConnectionSettingInfoMap</a> See documentation on <a href="#">IConnection::GetConnectionSettingInfo()</a>.</i>
virtual <a href="#">AttributeData</a> *	<b>GetCustomProperty</b> (simba_int32 in_key) <i>Retrieves a custom property value.</i>
virtual <a href="#">AttributeType</a>	<b>GetCustomPropertyType</b> (simba_int32 in_key) <i>Retrieves the type associated with the custom property with the given ODBC attribute key.</i>
virtual const <a href="#">simba_wstring</a> &	<b>GetDataSourceName</b> () <i>Retrieves the name of the data source.</i>
virtual const <a href="#">simba_string</a> &	<b>GetLocale</b> () <i>Retrieves the connection-wide locale.</i>
virtual <a href="#">ILogger</a> *	<b>GetLog</b> () <i>Gets the <a href="#">ILogger</a> for the connection.</i>

# Coding Standards



# Inheritance, Interfaces, Abstract Classes, Oh my!

## DSIRowCountResult Class Reference

An implementation of **IResult** which represents row count results. [More...](#)

Inherits **Simba::DSI::IResult**.

Inherited by **DSISimpleRowCountResult**.

[List of all members.](#)

### Public Member Functions

virtual simba_unsigned_native	<b>BulkFetch</b> (simba_unsigned_native in_rowsetSize, const std::vector< <b>Simba::DSI::IBulkProcessor</b> * > &in_bulkProcessors)
void	<b>CloseCursor</b> () <i>Closes the DSI's internal result cursor and clears associated memory.</i>
bool	<b>GetDataNeeded</b> (simba_uint16 in_column) <i>Gets the data needed status of a column as set by SetDataNeeded.</i>
<b>Simba::DSI::ResultType</b>	<b>GetResultType</b> () <i>Returns the type of the result.</i>
virtual simba_unsigned_native	<b>GetRowCount</b> ()=0 <i>Returns the row count.</i>
<b>Simba::DSI::IColumns</b> *	<b>GetSelectColumns</b> () <i>Retrieves an IColumns* which can provide access to column metadata for each columns in the result.</i>
virtual bool	<b>HasRowCount</b> ()=0 <i>Determine if the number of rows is known.</i>
virtual bool	<b>IsBulkFetchSupported</b> (std::set< simba_uint32 > &in_boundColumnIndex)
virtual <b>Simba::DSI::ICellmark</b> *	<b>MarkCell</b> (simba_uint16 in_column) <i>Returns a cellmark to the column of the row at which the cursor is positioned.</i>
bool	<b>Move</b> ( <b>Simba::DSI::DSIDirection</b> in_direction, simba_signed_native in_offset) <i>Traverses the result set.</i>
bool	<b>RetrieveData</b> (simba_uint16 in_column, <b>SqlData</b> *in_data, simba_signed_native in_offset, simba_signed_native in_maxSize) <i>Fills in in_data with a chunk of data for the given column in the current row.</i>

Inheritance,  
Interfaces,  
Abstract Classes,  
Oh my!

## DSISimpleRowCountResult Class Reference

Simple row count result that takes a known row count. [More...](#)

Inherits [Simba::DSI::DSIRowCountResult](#).

[List of all members.](#)

### Public Member Functions

	<a href="#">DSISimpleRowCountResult</a> (simba_unsigned_native in_rowCount) <i>Constructor.</i>
virtual simba_unsigned_native	<a href="#">GetRowCount</a> () <i>Returns the row count.</i>
virtual bool	<a href="#">HasRowCount</a> () <i>Determine if the number of rows in the result set is known.</i>
virtual void	<a href="#">SetRowCount</a> (simba_unsigned_native in_rowCount) <i>Set the row count.</i>
	<a href="#">~DSISimpleRowCountResult</a> () <i>Destructor.</i>

# Ctors and Dtors in Inheritance

## DSISimpleRowCountResult

```
DSISimpleRowCountResult ( simba_unsigned_native in_rowCount )
```

Constructor.

**Parameters:**  
*in\_rowCount* The row count for this result.

```
~DSISimpleRowCountResult ( )
```

Destructor.

## DSIRowCountResult

```
virtual ~DSIRowCountResult ( ) [virtual]
```

Destructor.

```
DSIRowCountResult ( ) [protected]
```

Constructor.

```
DSIRowCountResult* myRowCount = new DSISimpleRowCountResult(100);
```

## DSIMessageCache Class Reference

This class acts as a cache for all warning and error messages loaded through the DSII. [More...](#)

[List of all members.](#)

### Public Member Functions

```
void AddErrorMessage (const simba_string &in_locale, const simba_wstring &in_messageID, simba_int32 in_sourceComponentID, const simba_wstring &in_message, simba_int32 in_nativeErrCode)  
Adds the specified message, keyed with the message ID and component ID.
```

```
DSIMessageCache ()  
Constructor.
```

```
bool GetErrorMessage (const simba_string &in_locale, const simba_wstring &in_messageID, simba_int32 in_sourceComponentID, simba_wstring &out_message, simba_int32 &out_nativeErrCode)  
Retrieves the message specified by the message ID and component ID.
```

```
~DSIMessageCache ()  
Destructor.
```

### Detailed Description

This class acts as a cache for all warning and error messages loaded through the DSII.

### Constructor & Destructor Documentation

```
DSIMessageCache ( )
```

Constructor.

```
~DSIMessageCache ( )
```

Destructor.

### Member Function Documentation

```
void AddErrorMessage ( const simba_string & in_locale,  
                      const simba_wstring & in_messageID,  
                      simba_int32 in_sourceComponentID,  
                      const simba_wstring & in_message,  
                      simba_int32 in_nativeErrCode  
                      )
```

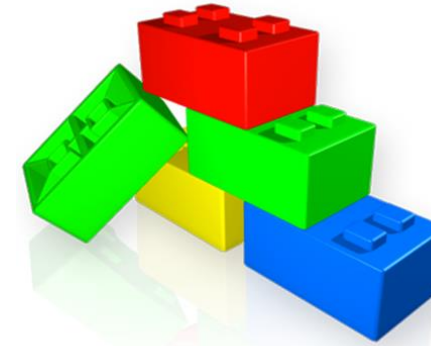
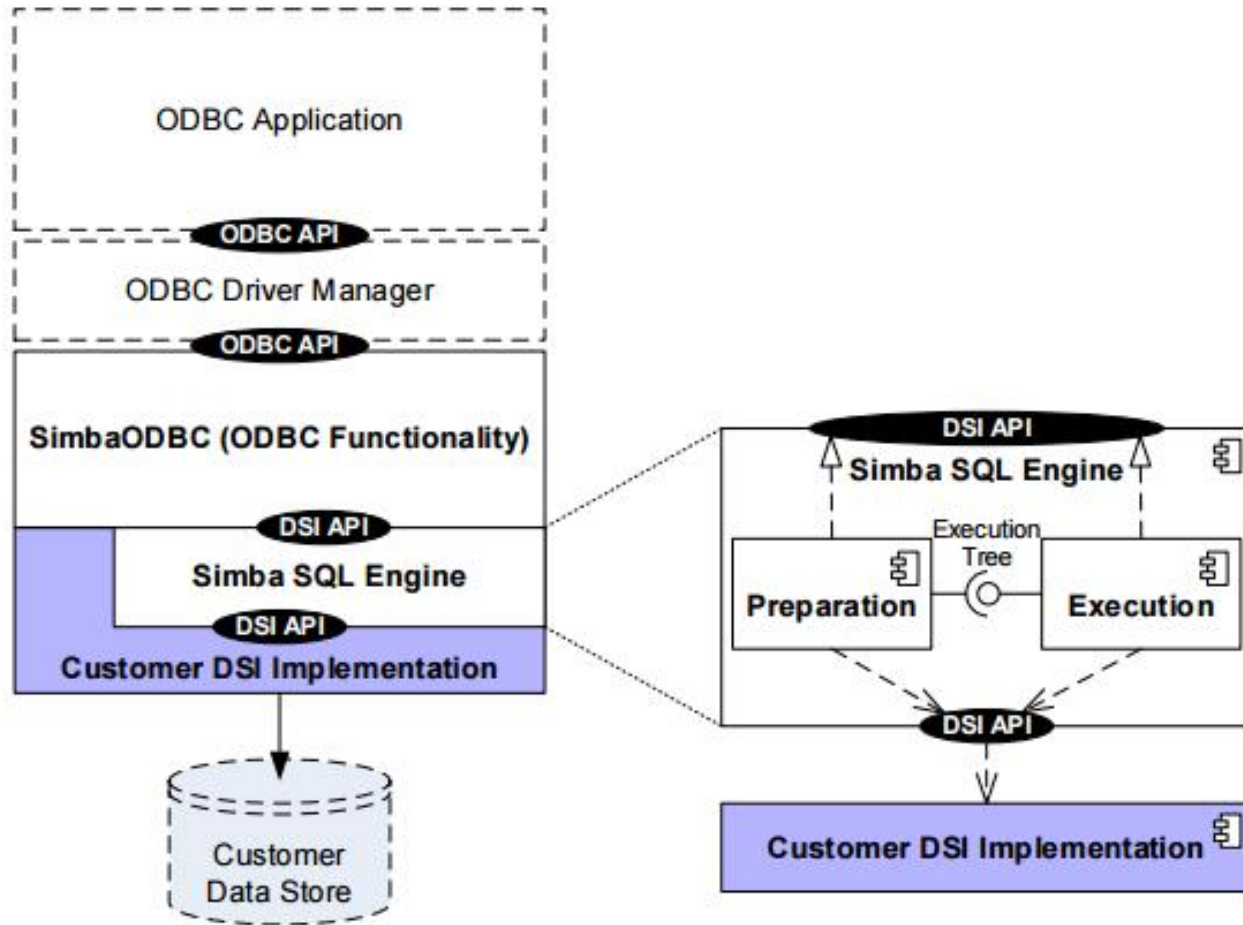
Adds the specified message, keyed with the message ID and component ID.

If an existing message is stored for the two IDs, it will not be overwritten.

#### Parameters:

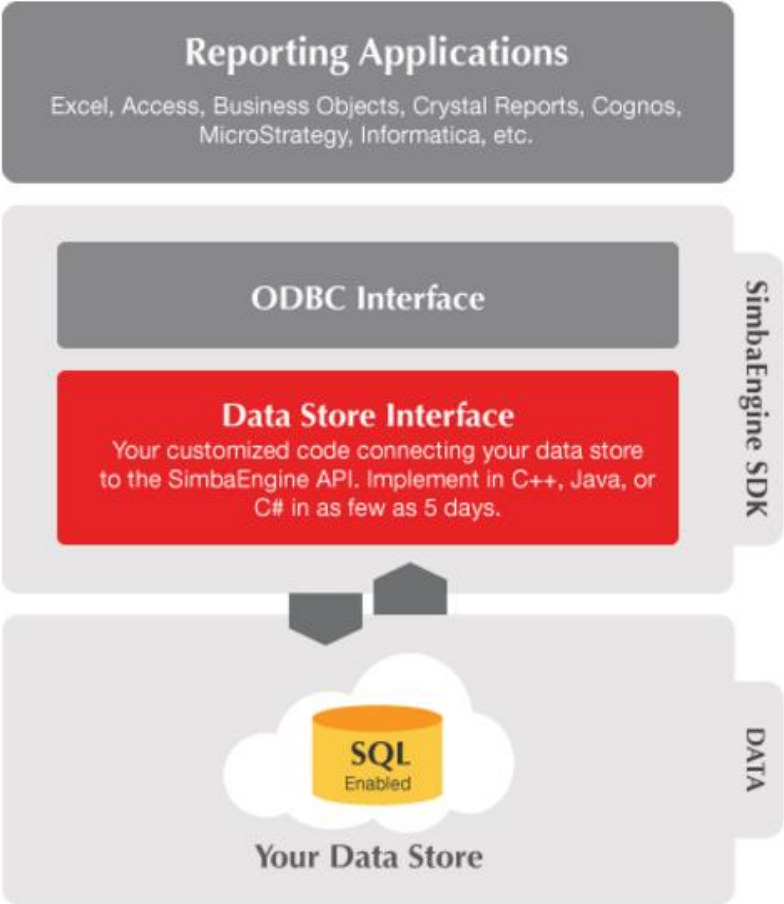
<i>in_locale</i>	The locale.
<i>in_messageID</i>	Unique message identifier.

# Don't Break Encapsulation

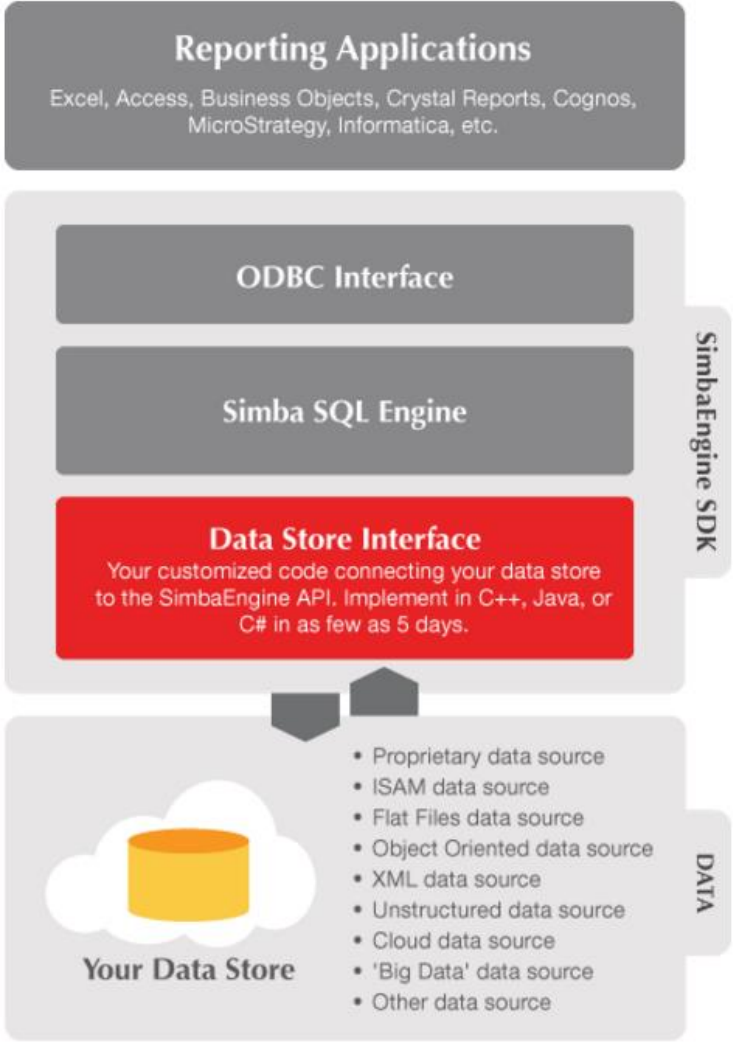


# Modularization

# If your data source is SQL-enabled



# If your data source is not SQL-enabled



# Simba SQL Engine

- Comprised of the following components:
  - **Core**
    - Classes common to all the other SQL Engine projects.
  - **Parser**
    - Parses incoming SQL queries and builds Algebraic Expression Trees (AETree).
  - **AEProcessor**
    - Builds, processes and optimizes the abstract algebraic expressions that result from the parsing of a SQL query.
  - **Executor**
    - Contains execution trees which are generated from AETrees, and is responsible for the actual execution of a query.
  - **DSIExt**
    - An extension of the DSI, this project acts as an interface the DSII can implement to leverage the SQL Engine.

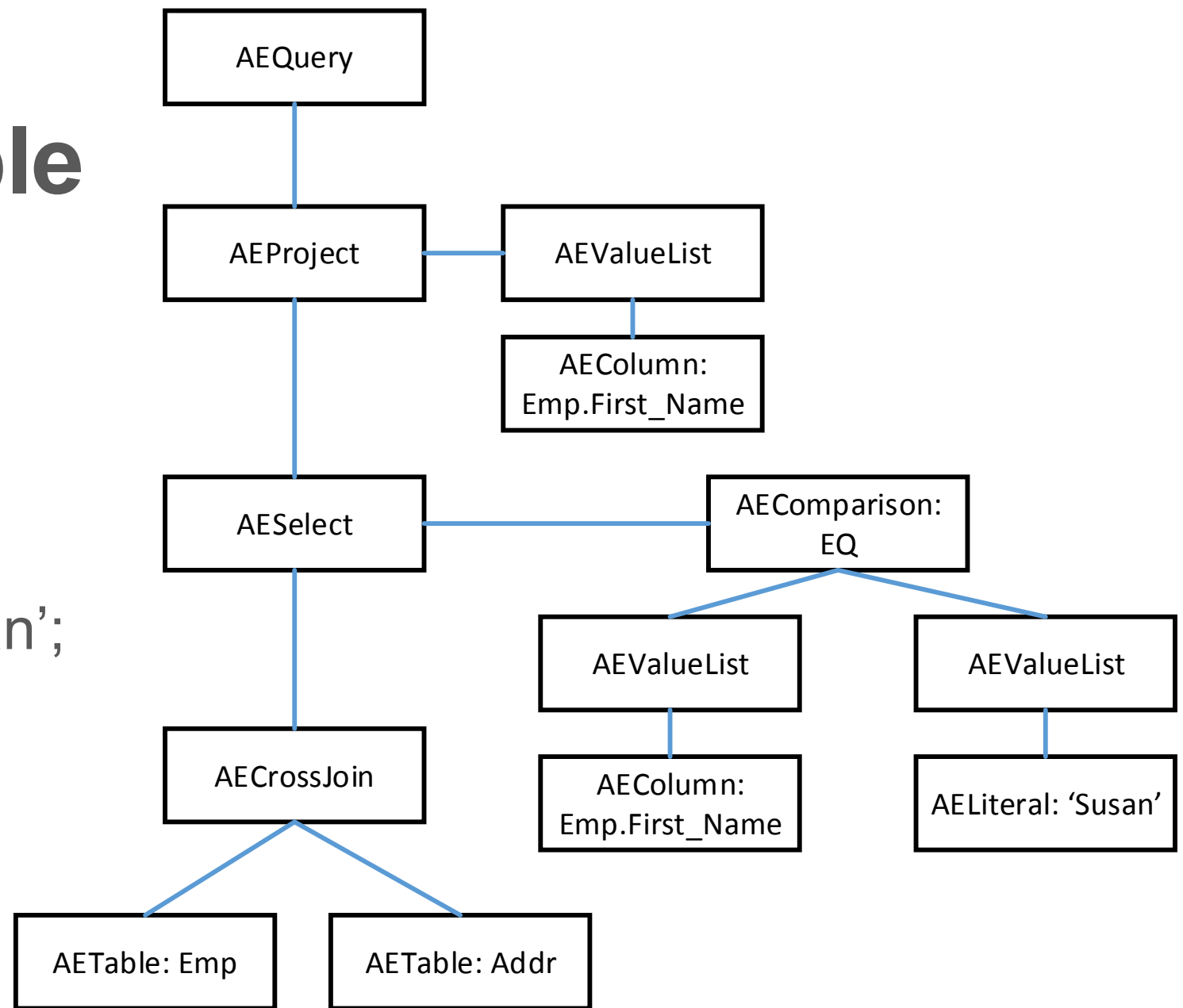
# Algebraic Expression Trees (AETrees)

- Are a component in the AEProcessor
- Algebraic Expression representation of the SQL query
- Intermediary representation between Parse Tree and Execution Tree



# AETree Example

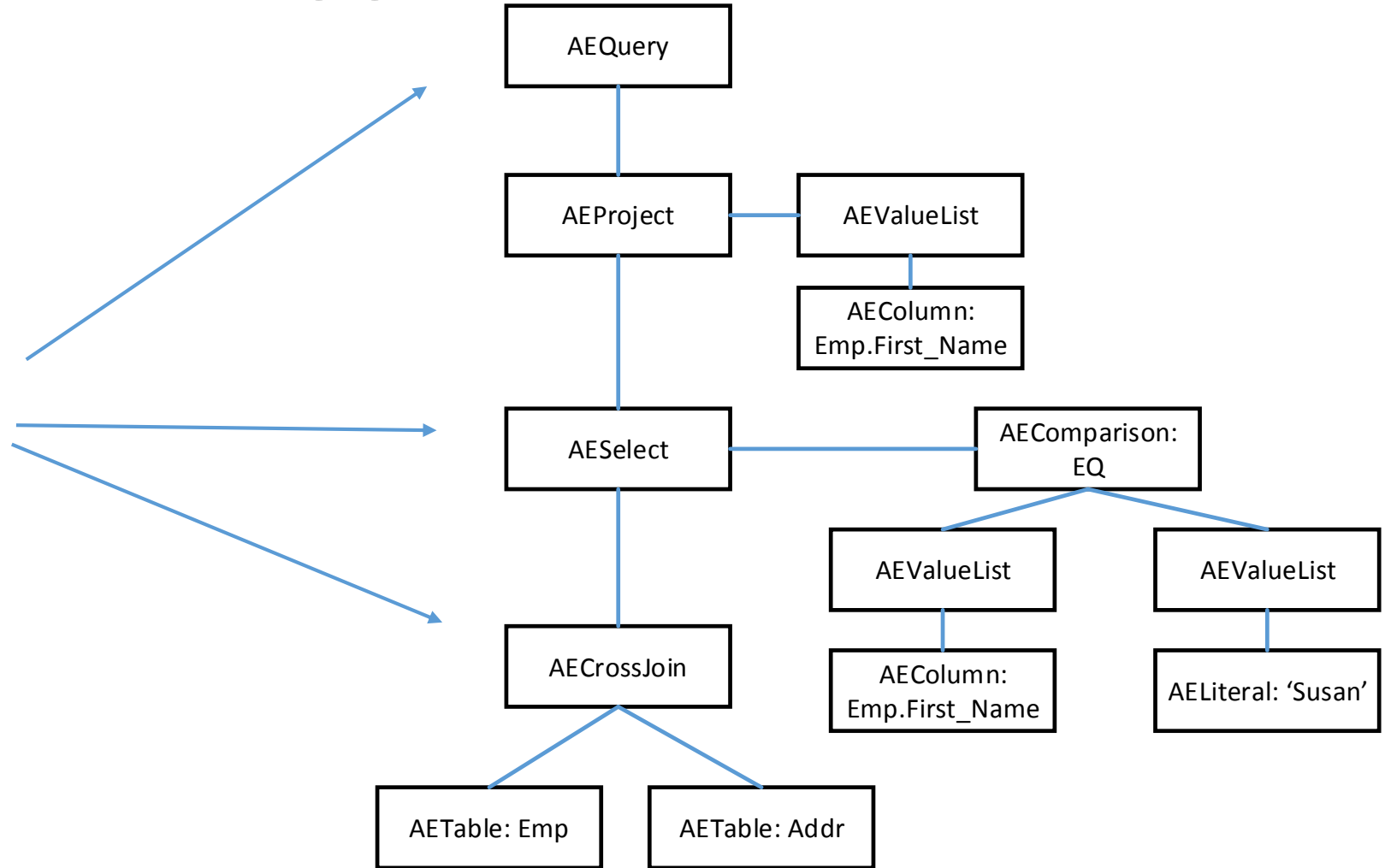
```
SELECT Emp.first_name  
FROM EMP, ADDR  
WHERE  
EMP.FIRST_NAME = 'Susan';
```



# Let's Design AETree

These are all nodes...

**AENode**



## AENode Member List

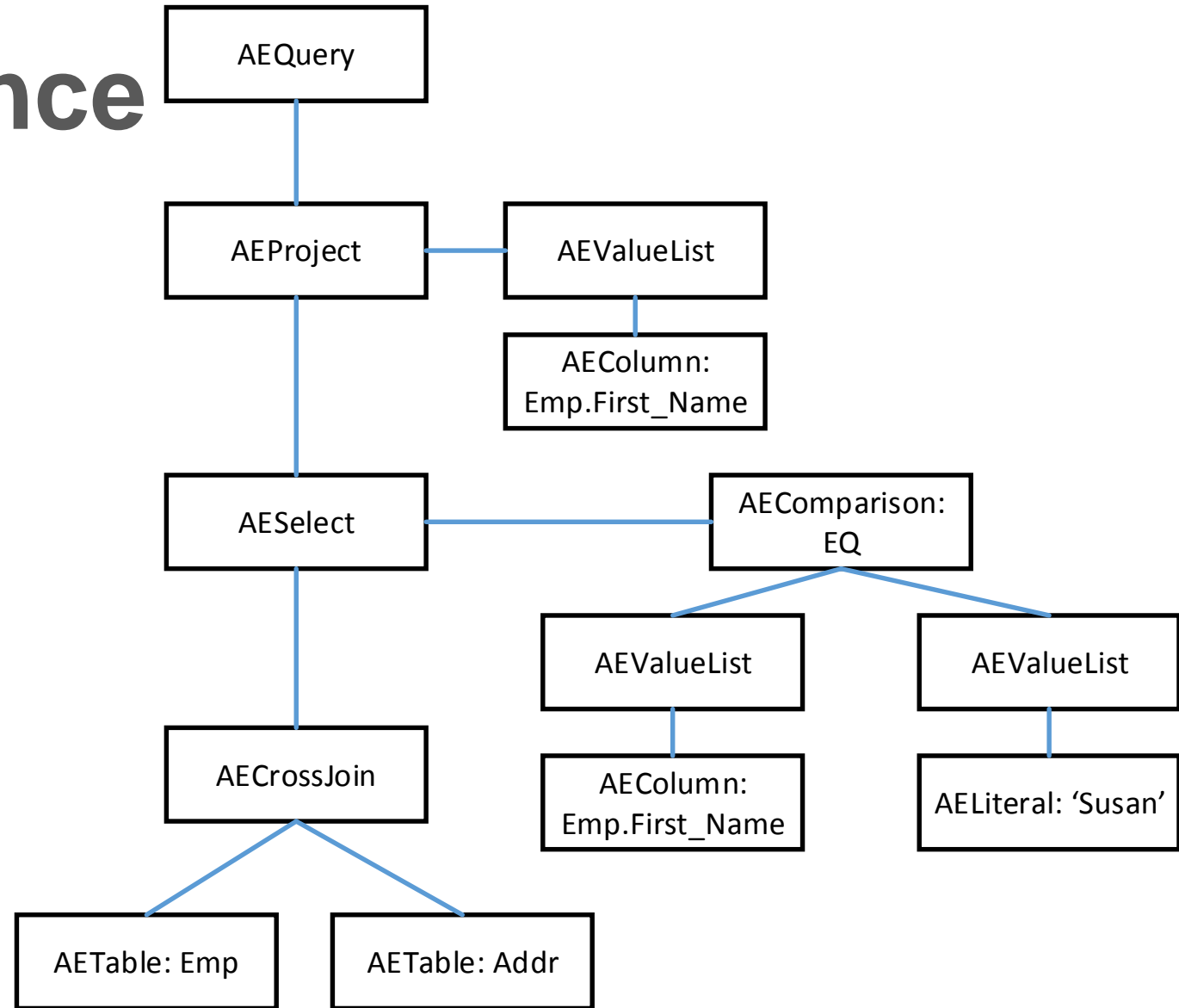
This is the complete list of members for **AENode**, including all inherited members.

<b>AcceptVisitor</b> (AENodeVisitor &in_visitor)=0	<b>AENode</b> [pure virtual]
<b>AENode</b> ()	<b>AENode</b> [protected]
<b>AENode</b> (const AENode &in_other)	<b>AENode</b> [protected]
<b>Clone</b> () const =0	<b>AENode</b> [pure virtual]
<b>GetAsBooleanExpr</b> ()	<b>AENode</b> [virtual]
<b>GetAsBooleanExpr</b> () const	<b>AENode</b> [virtual]
<b>GetAsQueryOperation</b> ()	<b>AENode</b> [virtual]
<b>GetAsQueryOperation</b> () const	<b>AENode</b> [virtual]
<b>GetAsRelationalExpr</b> ()	<b>AENode</b> [virtual]
<b>GetAsRelationalExpr</b> () const	<b>AENode</b> [virtual]
<b>GetAsSetClause</b> ()	<b>AENode</b> [virtual]
<b>GetAsSetClause</b> () const	<b>AENode</b> [virtual]
<b>GetAsSetClauseList</b> ()	<b>AENode</b> [virtual]
<b>GetAsSetClauseList</b> () const	<b>AENode</b> [virtual]
<b>GetAsStatement</b> ()	<b>AENode</b> [virtual]
<b>GetAsStatement</b> () const	<b>AENode</b> [virtual]
<b>GetAsValueExpr</b> ()	<b>AENode</b> [virtual]
<b>GetAsValueExpr</b> () const	<b>AENode</b> [virtual]
<b>GetAsValueList</b> ()	<b>AENode</b> [virtual]
<b>GetAsValueList</b> () const	<b>AENode</b> [virtual]
<b>GetChild</b> (simba_size_t in_index)=0	<b>AENode</b> [pure virtual]
<b>GetChild</b> (simba_size_t in_index) const =0	<b>AENode</b> [pure virtual]
<b>GetChildCount</b> () const =0	<b>AENode</b> [pure virtual]
<b>GetChildren</b> ()	<b>AENode</b>
<b>GetLogString</b> () const =0	<b>AENode</b> [pure virtual]
<b>GetNodeType</b> () const =0	<b>AENode</b> [pure virtual]
<b>GetParent</b> ()	<b>AENode</b>
<b>GetRoot</b> ()	<b>AENode</b>
<b>IsBooleanExpr</b> () const	<b>AENode</b> [virtual]
<b>IsEqual</b> (const AENode *in_another) const	<b>AENode</b> [protected, virtual]
<b>IsEquivalent</b> (const AENode *in_another) const	<b>AENode</b> [virtual]
<b>IsQueryOperation</b> () const	<b>AENode</b> [virtual]
<b>IsRelationalExpr</b> () const	<b>AENode</b> [virtual]
<b>IsStatement</b> () const	<b>AENode</b> [virtual]
<b>IsValueExpr</b> () const	<b>AENode</b> [virtual]
<b>SetParent</b> (AENode *in_node)	<b>AENode</b>
<b>Validate</b> ()	<b>AENode</b> [virtual]
<b>~AENode</b> ()	<b>AENode</b> [virtual]

# AENode Class

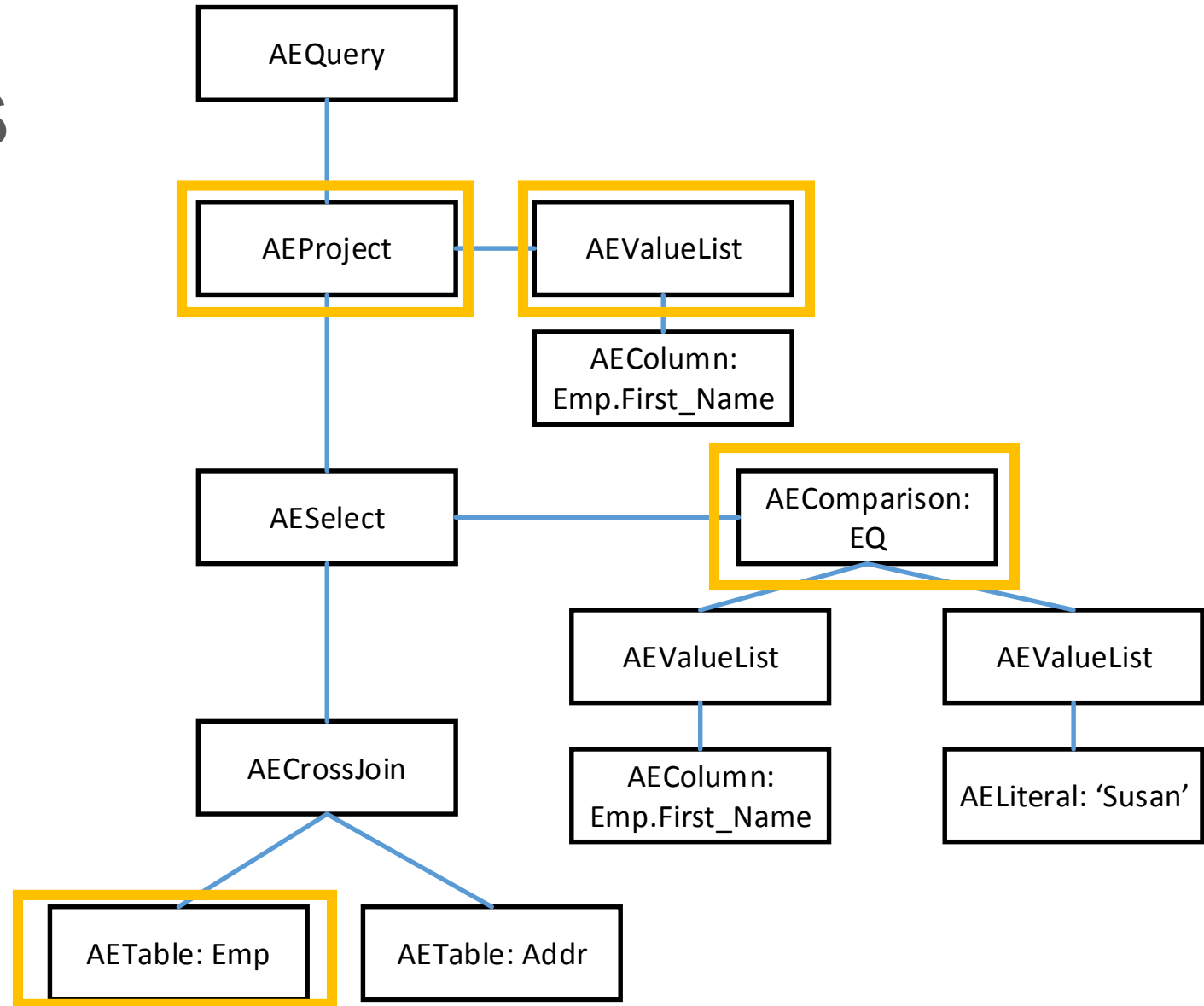
# Applying Inheritance

- Categorize the types of nodes
  - Boolean Expressions: AECOMPARISON
  - Value Expressions: AECOLUMN, AELITERAL
  - Relational Expressions: AEPROJECT, Aeselect, AECROSSJOIN, AETABLE



# Using Templates

- A different type of categorization
- Instead of the types of nodes, think about the structure of a tree made up of nodes
  - Terminal (leaf) nodes
  - Parent nodes w/ 1 child
  - Parent nodes w/ 2 children
  - Parent nodes w/ 1 ... many children



# Terminal (Leaf) Nodes: AETerminalExprT

## AETerminalExprT< BaseNodeT > Class Template Reference

A template class that models an operation that takes no operand. [More...](#)

[List of all members.](#)

### Public Member Functions

virtual const <b>AENode</b> *	<b>GetChild</b> (simba_size_t in_index) const <i>Returns the specified child node.</i>
virtual <b>AENode</b> *	<b>GetChild</b> (simba_size_t in_index) <i>Returns the specified child node.</i>
virtual simba_size_t	<b>GetChildCount</b> () const <i>Returns the number of child nodes. In this case, it always return 0.</i>
virtual	<b>~AETerminalExprT</b> () <i>Destructor.</i>

### Protected Member Functions

	<b>AETerminalExprT</b> (const <b>AETerminalExprT</b> &in_other) <i>Copy constructor.</i>
	<b>AETerminalExprT</b> () <i>Constructor.</i>

# Parent Nodes w/ 1 child: AEUnaryExprT

## AEUnaryExprT< BaseNodeT, OperandT, OperandPtrT > Class Template Reference

A template class that models an operation that takes only one operand. [More...](#)

[List of all members.](#)

### Public Member Functions

virtual const <b>AENode</b>	* <b>GetChild</b> (simba_size_t in_index) const <i>Returns the specified child node.</i>
virtual <b>AENode</b> *	<b>GetChild</b> (simba_size_t in_index) <i>Returns the specified child node.</i>
virtual simba_size_t	<b>GetChildCount</b> () const <i>Returns the number of child nodes. In this case, it always return 1.</i>
const OperandT *	<b>GetOperand</b> () const <i>Retrieves a pointer reference to the only operand that this unary operation operates on.</i>
OperandT *	<b>GetOperand</b> () <i>Retrieves a pointer reference to the only operand that this unary operation operates on.</i>
void	<b>SetOperand</b> (OperandPtrT in_operand) <i>Sets the only operand that this unary operation operates on.</i>
OperandPtrT	<b>TakeOperand</b> () <i>Takes the ownership of the operand away from this object.</i>
virtual	<b>~AEUnaryExprT</b> () <i>Destructor.</i>

### Protected Member Functions

	<b>AEUnaryExprT</b> (const <b>AEUnaryExprT</b> &in_other) <i>Copy constructor.</i>
	<b>AEUnaryExprT</b> (OperandPtrT in_operand) <i>Constructor.</i>
	<b>AEUnaryExprT</b> () <i>Default constructor.</i>

### Protected Attributes

OperandPtrT	<b>m_operand</b>
-------------	------------------

# AEBinaryExprT< BaseNodeT, LOperandT, ROperandT, LOperandPtrT, ROperandPtrT > Class Template Reference

A template class that models an operation that takes two operands. [More...](#)

Inherited by [AEBinaryValueExpr](#), [AELikePredicate](#), [AEQuantifiedComparison](#), and [AESimpleWhenClause](#).

[List of all members.](#)

## Public Member Functions

virtual const	<b>AENode *</b> <a href="#">GetChild</a> (simba_size_t in_index) const <i>Returns the specified child node.</i>
virtual <b>AENode *</b>	<a href="#">GetChild</a> (simba_size_t in_index) <i>Returns the specified child node.</i>
virtual simba_size_t	<b>GetChildCount</b> () const <i>Returns the number of child nodes. In this case, it always returns 2.</i>
const LOperandT *	<b>GetLeftOperand</b> () const <i>Returns a pointer reference to the left operand.</i>
LOperandT *	<a href="#">GetLeftOperand</a> () <i>Returns a pointer reference to the left operand.</i>
const ROperandT *	<b>GetRightOperand</b> () const <i>Returns a pointer reference to the right operand.</i>
ROperandT *	<a href="#">GetRightOperand</a> () <i>Returns a pointer reference to the right operand.</i>
void	<b>SetLeftOperand</b> (LOperandPtrT in_leftOperand) <i>Grants the ownership of the given object wrapped in an auto pointer to this object as the left operand.</i>
void	<b>SetRightOperand</b> (ROperandPtrT in_rightOperand) <i>Grants the ownership of the given object wrapped in an auto pointer to this object as the right operand.</i>
LOperandPtrT	<b>TakeLeftOperand</b> () <i>Takes the ownership of the left operand away from this object.</i>
ROperandPtrT	<b>TakeRightOperand</b> () <i>Takes the ownership of the right operand away from this object.</i>
virtual	<b>~AEBinaryExprT</b> () <i>Destructor.</i>

## Protected Member Functions

	<b>AEBinaryExprT</b> (const <b>AEBinaryExprT</b> &in_other) <i>Copy constructor.</i>
	<b>AEBinaryExprT</b> (LOperandPtrT in_leftOperand, ROperandPtrT in_rightOperand) <i>Constructor.</i>
	<b>AEBinaryExprT</b> () <i>Constructor.</i>

## Protected Attributes

LOperandPtrT	<b>m_leftOperand</b> <i>The operand on the left side of the binary operation. (OWN).</i>
ROperandPtrT	<b>m_rightOperand</b> <i>The operand on the right side of the binary operation. (OWN).</i>

# Parent Nodes w/ 2 children: AEBinaryExprT



## AENodeListT< BaseNodeT, ItemNodeT > Class Template Reference

A template class that holds a list of nodes of type ItemNodeT managed by a shared pointer. [More...](#)

[List of all members.](#)

### Public Member Functions

simba_size_t	<b>AddNode</b> (SharedItemNodeT in_node) <i>Appends the given node to the end of the list.</i>
	<b>AENodeListT</b> (const AENodeListT &in_other) <i>Copy constructor.</i>
	<b>AENodeListT</b> () <i>Default constructor.</i>
bool	<b>FindNode</b> (ItemNodeT *in_node, simba_size_t in_startPos, simba_size_t in_endPos, simba_size_t &out_pos) <i>Finds the index of the matching node.</i>
bool	<b>FindNode</b> (ItemNodeT *in_node, simba_size_t &out_pos) <i>Finds the index of the matching node.</i>
virtual const ItemNodeT *	<b>GetChild</b> (simba_size_t in_index) const <i>Gets the node indexed by the given node number.</i>
virtual ItemNodeT *	<b>GetChild</b> (simba_size_t in_index) <i>Gets the node indexed by the given node number.</i>
virtual simba_size_t	<b>GetChildCount</b> () const <i>Gets the number of child nodes.</i>
SharedItemNodeT	<b>ReplaceNode</b> (simba_size_t in_index, SharedItemNodeT in_node) <i>Replaces the node at the given index with the given node.</i>
virtual	<b>~AENodeListT</b> () <i>Destructor.</i>

Parent nodes w/ 1  
... many children:  
AENodeListT

# Summary

- Identify the use case
- Identify the customer
- Know thy customer
- Write enterprise-level code
  - What owns what?
  - Is this class derivable?
  - Do I need an interface?
  - Think about objects in different ways – inheritance AND templates?

# Simba is hiring



This is the view  
from our office,  
by the way



# Resources

- Simba's website: <http://www.simba.com/>
- Documentation for Simba Engine SDK: [http://www.simba.com/products/simba-engine-sdk#documentation\\_content](http://www.simba.com/products/simba-engine-sdk#documentation_content)
- Careers at Simba: <https://careers-simba.icims.com/jobs/>

