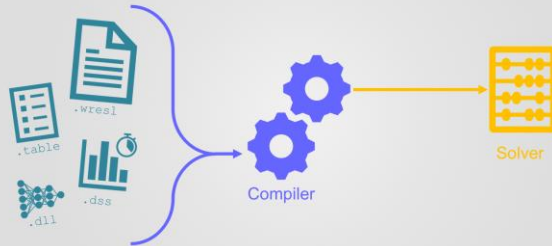


WRESL+ Compiler Updates



Agenda

What is the WRESL+ Compiler

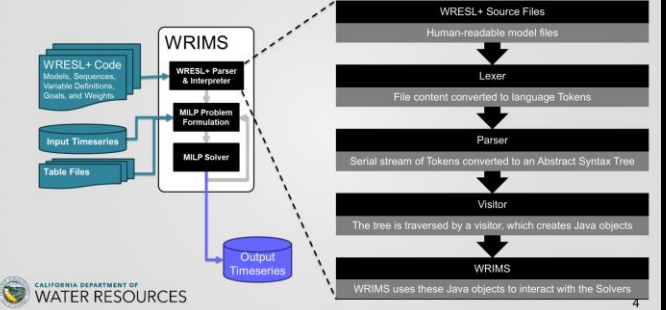


Project Objectives

- Primary Goal**
 - Improve maintainability of WRESL+
 - Add new features to WRESL+
 - Allow for non-Java uses of the compiler tools
- Background**
 - ANTLR automatically writes compiler code given a "Grammar"
- Why Now?**
 - ANTLR version 3 is no longer maintained.
 - ANTLR 4 has appealing features.
- Key Problem**
 - **Current Grammar file is not compatible with ANTLR 4.**
- Our Plan**
 - Rewrite grammar files in ANTLR 4 style
 - Define the grammar without ties to Java
 - Rewrite the Java parsing toolchain in WRIMS to take advantage of ANTLR 4 features.



Overview of Compiler Workflow



Danger of Embedded Actions

WRESL+ "Model" syntax definition in ANTLR 3

```

157 model
158 {
159   @init { String id = null; }
160   : MODEL i=IDENT '(' (pattern)+ ')'
161   { id = $i.text.toLowerCase();
162     model_list.add(id);
163     -> { model_in_sequence.contains(id)? ^(MODEL IDENT[id] (pattern)+ ) :
164       // ignore
165   } ;

```

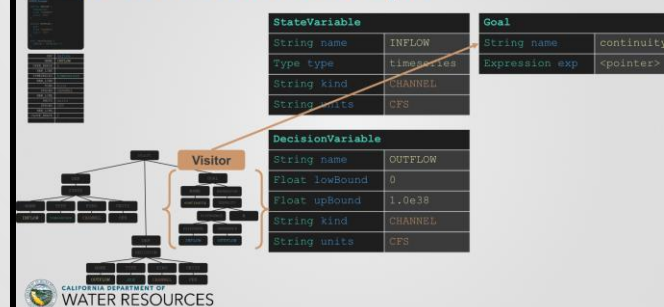
Annotations in the code:

- Java variable declaration: `String id = null;`
- Java string manipulation: `id = $i.text.toLowerCase();`
- Java object mutation: `model_list.add(id);`
- Java condition check: `model_in_sequence.contains(id)?`

Highlighting the embedded Java actions that make the grammar file unusable by other languages



Visitor Pattern Example



Summary

- Modern**
 - Migrating from ANTLR 3 to 4 brings the WRESL+ compiler onto an actively maintained, well-supported foundation
- Architectural Improvements**
 - Separation of grammar and lexer/parser improves maintainability
- Future Ready**
 - New language features and better tooling are now possible
- Better User Experience**
 - Clearer error messages, and modern IDE support are on the way



What is WRESL+?

Domain Specific

- Developed by DWR.
- Used in WRIMS.

It's a Translator

- Bridge between water domain concepts and optimization domain concepts.

```
goal EI_split_swp {  
  lhs    C_CAA003_EXP1_TD  
  
  ! Only apply goal is EI_control_on switch is set to "1"  
  case ei_control {  
    condition EI_control_on > 0.5  
  
    rhs      (SWPExpCapShare_Blncd * EiExpCtrl[PRESETUP])  
             + (ExpRatio * C_FTR059_m)  
             - (ExpRatio * C_FTR059_m[PRESETUP])  
  
    lhs>rhs  penalty 1285  
    lhs<rhs  penalty 0  
  } ! end case ei_control  
} ! end goal
```



Project Objectives

Primary Goals

1. Improve maintainability of WRESL+
2. Add new features to WRESL+
3. Allow for non-Java uses of the compiler tools

Background

ANTLR automatically writes compiler code given a “grammar”

Why Now?

*ANTLR 3 is no longer maintained.
ANTLR 4 has appealing features.*

Key Problem

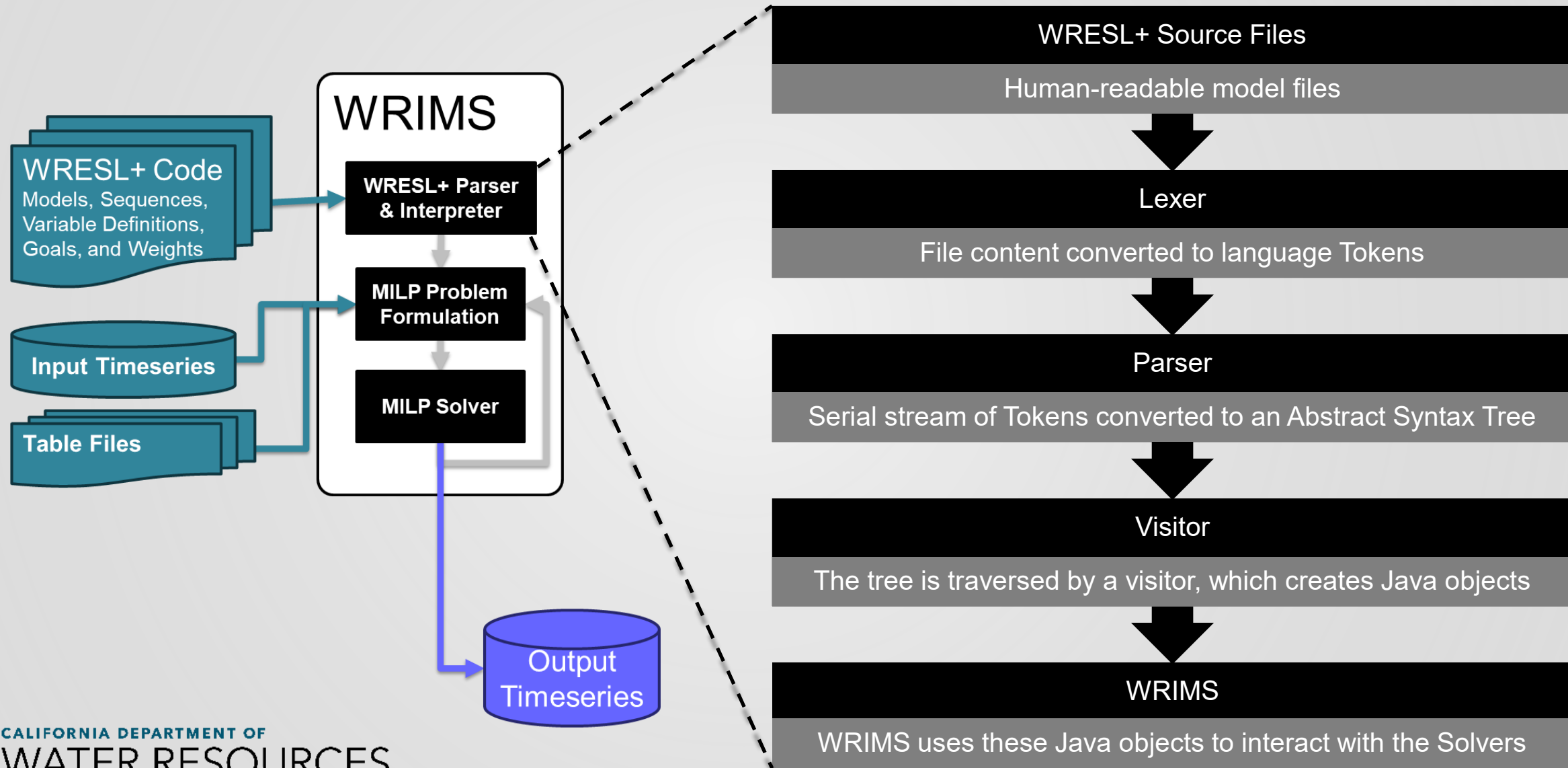
Current grammar file is not compatible with ANTLR 4.

Our Plan

1. Rewrite grammar files in ANTLR 4 style
2. Define the grammar without ties to Java
3. Rewrite the Java parsing toolchain in WRIMS to take advantage of ANTLR 4 features.



Overview of Compiler Workflow



Danger of Embedded Actions

WRESL+ “Model” syntax definition in ANTLR 3

```
157
158 model
159 @init { String id = null;}
160 : MODEL i=IDENT '{' (pattern )+ '}'
161     {id = $i.text.toLowerCase();
162     model_list.add(id);}
163 -> {model_in_sequence.contains(id)}? ^(Model IDENT[id] (pattern )+ )
164 -> // ignore
165 ;
```



Danger of Embedded Actions

WRESL+ “Model” syntax definition in ANTLR 3

```
157
158 model
159 @init { String id = null;}
160 : MODEL i=IDENT '{' (pattern )+ '}'
161     {id = $i.text.toLowerCase();
162     model_list.add(id);}
163 -> {model_in_sequence.contains(id)}? ^(Model IDENT[id] (pattern )+ )
164 -> // ignore
165 ;
```

Highlighting only the “grammar” definition, everything else is an embedded action.



Danger of Embedded Actions

WRESL+ “Model” syntax definition in ANTLR 3

```
157
158 model
159 @init { String id = null;
160       : MODEL i=IDENT '{' (pattern )+ '}'
161       { id = $i.text.toLowerCase();
162         model_list.add(id);
163       -> {model_in_sequence.contains(id)}? ^(Model IDENT[id] (pattern )+ )
164       -> // ignore
165       ;
```

Java variable declaration

Java string manipulation

Java object mutation

Java condition check

Highlighting the embedded Java actions that make the grammar file unusable by other languages



Transition Technical Improvements

ANTLR 3

Multiple Grammar files

Embedded Actions in Grammar

Grammar locked to Java

ANTLR 4

Single Grammar File

Visitor pattern used
(Grammar is static)

Language agnostic grammar
(Java compiler for WRIMS)



Visitor Pattern Example

WRESL Example

```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

- How does ANTLR process this text?
 1. Lexer
 - Breaks up the text into tokens
 2. Parser
 - Creates Abstract Syntax Tree
 3. Visitor
 - Builds model objects



Visitor Pattern Example



WRESL Example

```
define INFLOW {
    timeseries
    kind 'CHANNEL'
    units 'CFS'
}

define OUTFLOW {
    std
    kind 'CHANNEL'
    units 'CFS'
}

goal continuity {
    INFLOW - OUTFLOW = 0
}
```

[illegible]

Visitor Pattern Example

Lexer

Parser

Visitor

WRESL Example

```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

Token Stream

DEF	define
NAME	INFLOW
...	...



Visitor Pattern Example

Lexer

Parser

Visitor

WRESL Example

```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

Token Stream

DEF	define
NAME	INFLOW
OPEN_BRACE	{
...	...



Visitor Pattern Example

Lexer

Parser

Visitor

WRESL Example

```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

Token Stream

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
...	...



Visitor Pattern Example

Lexer

Parser

Visitor

WRESL Example

```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

Token Stream

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
...	...



Visitor Pattern Example

Lexer

Parser

Visitor

WRESL Example

```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

Token Stream

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
...	...



Visitor Pattern Example

Lexer

Parser

Visitor

WRESL Example

```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

Token Stream

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
...	...



Visitor Pattern Example

Lexer

Parser

Visitor

WRESL Example

```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

Token Stream

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
...	...



Visitor Pattern Example

Lexer

Parser

Visitor

WRESL Example

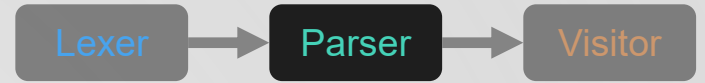
```
define INFLOW {  
  timeseries  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
define OUTFLOW {  
  std  
  kind 'CHANNEL'  
  units 'CFS'  
}  
  
goal continuity {  
  INFLOW - OUTFLOW = 0  
}
```

Token Stream

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}
...	...



Visitor Pattern Example



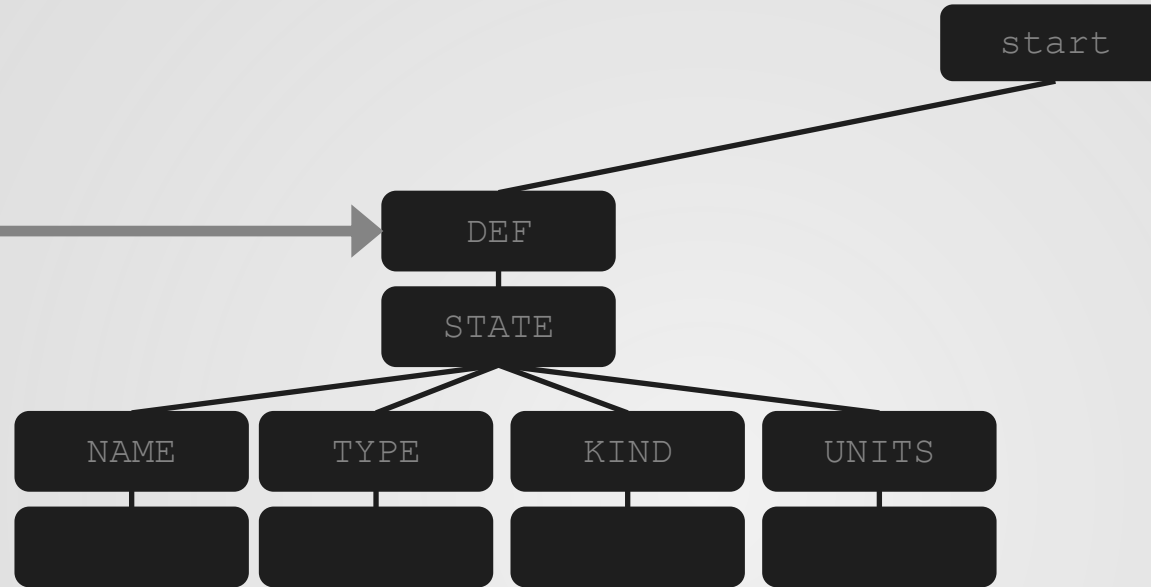
```

WRESL Example
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

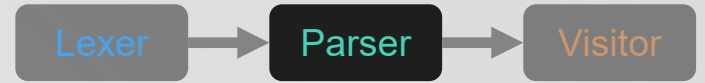
define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
  
```

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}
...	...



Visitor Pattern Example



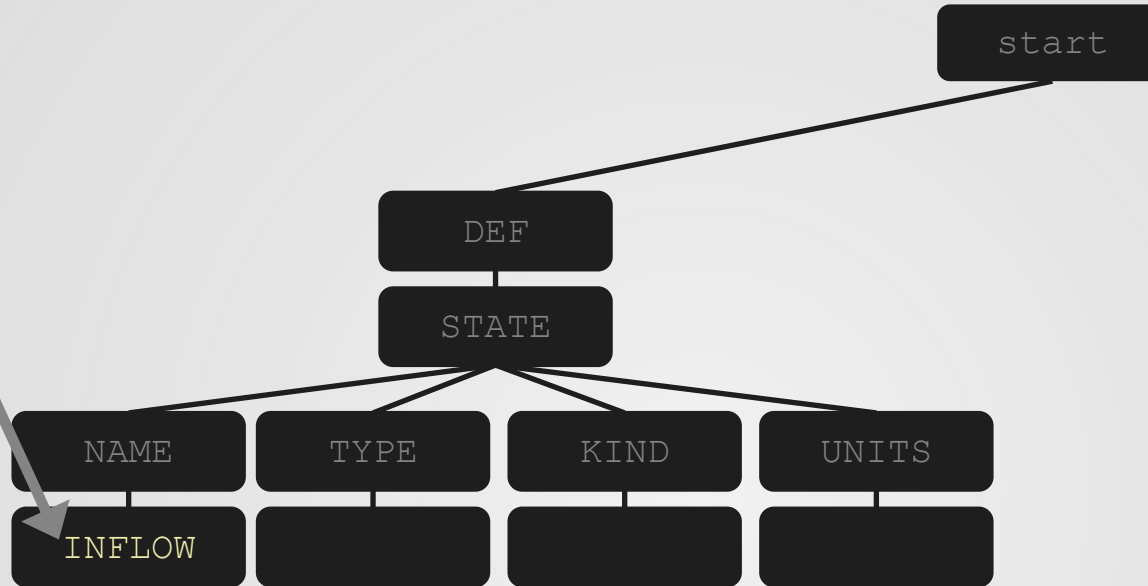
```

WRESL Example
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

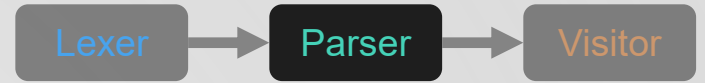
define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
  
```

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}
...	...



Visitor Pattern Example



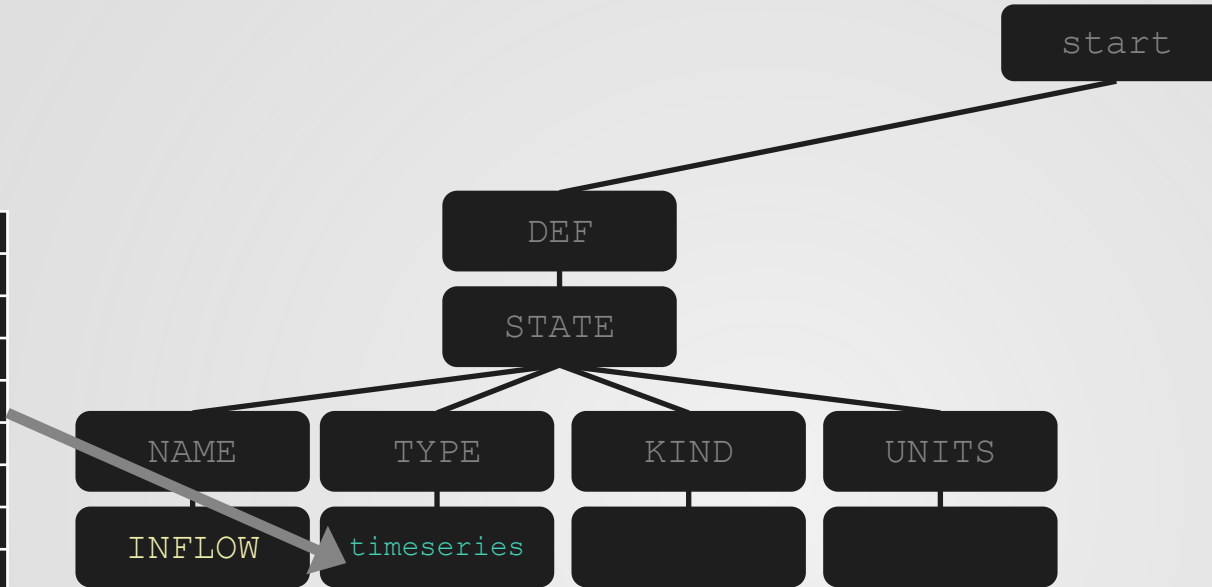
```

WRESL Example
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

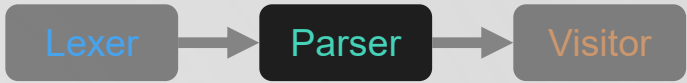
define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
  
```

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}
...	...



Visitor Pattern Example

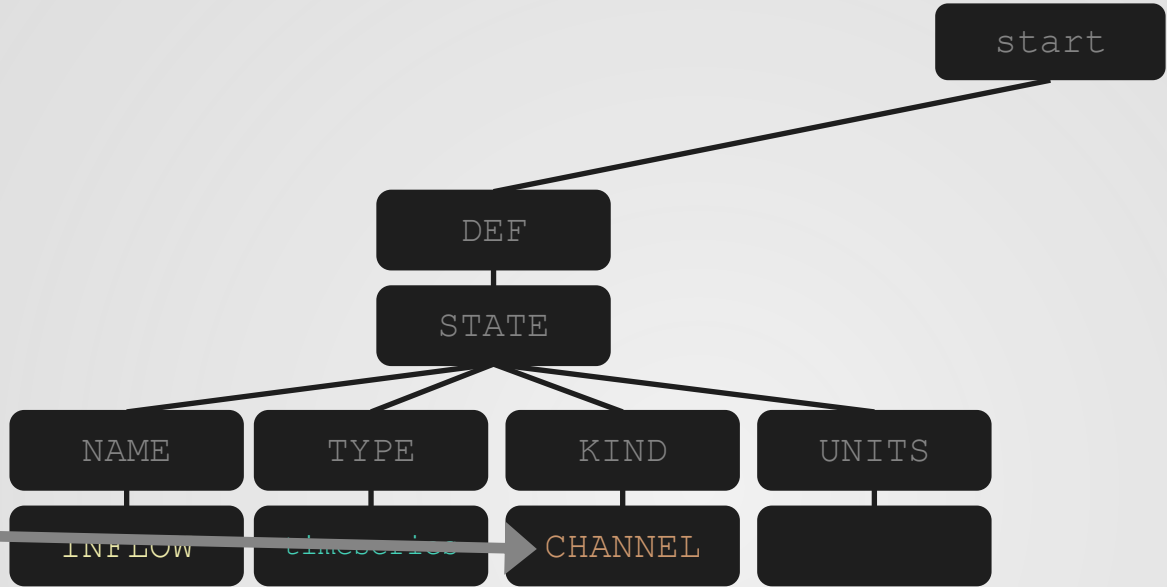


```
WRESL Example
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

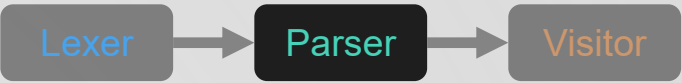
define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
```

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}
...	...



Visitor Pattern Example



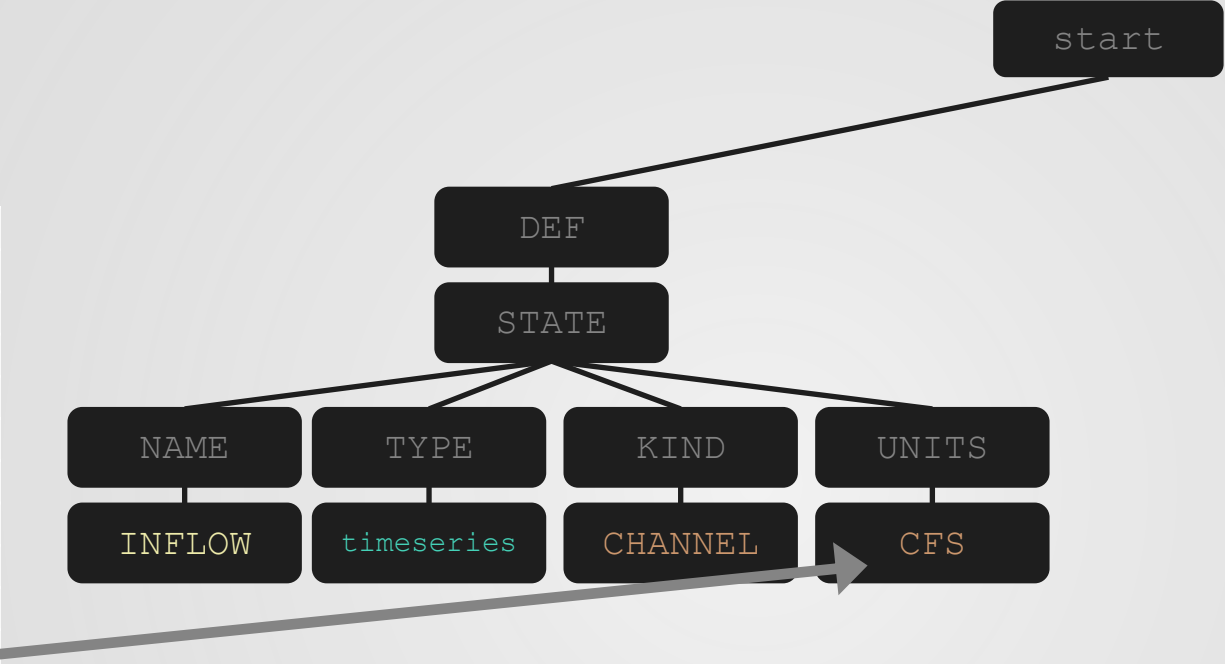
```

WRESL Example
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

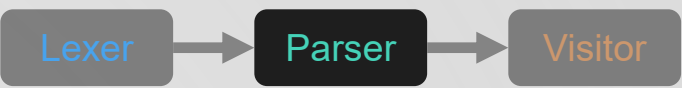
define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
  
```

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}
...	...



Visitor Pattern Example



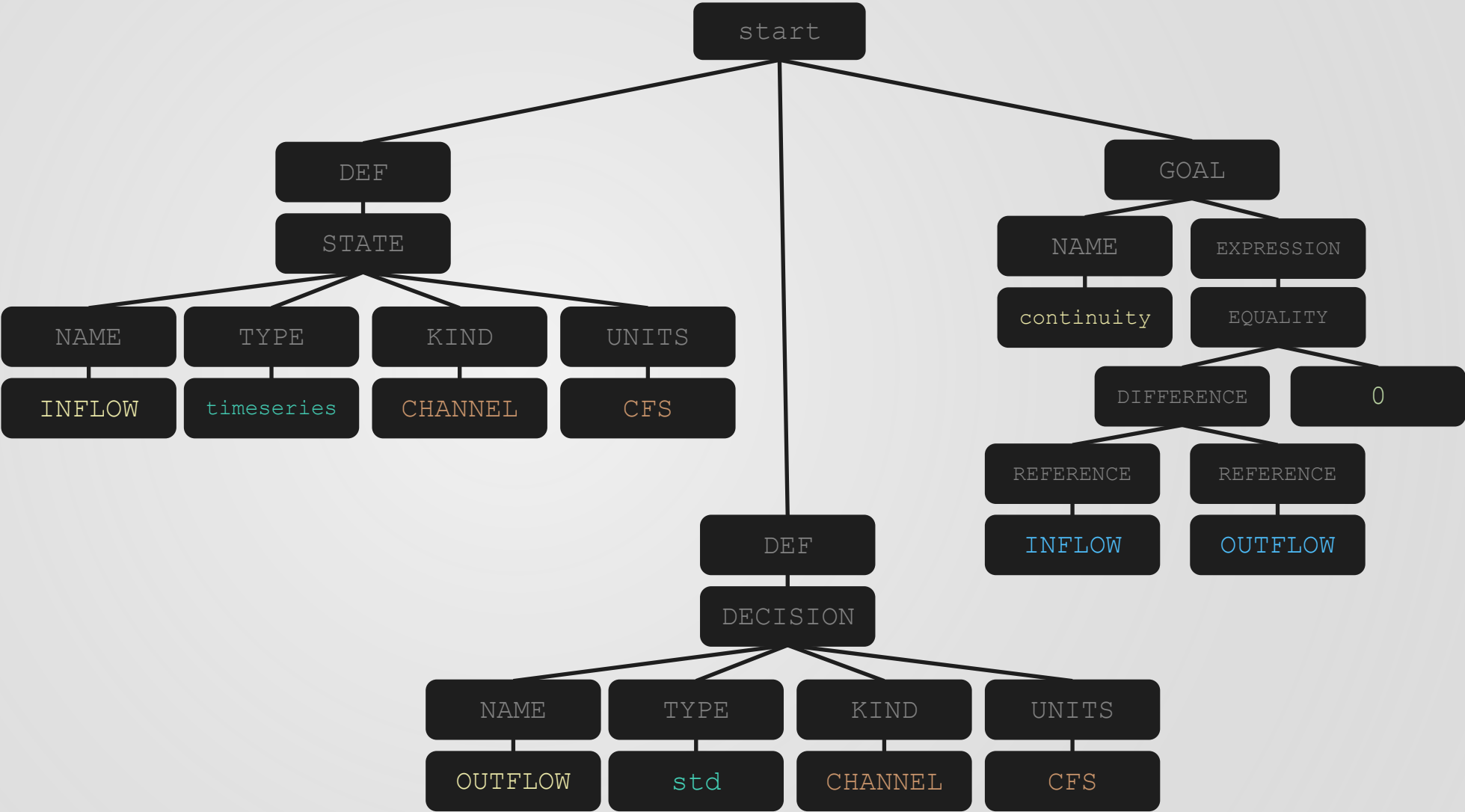
```

WRESL Example
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

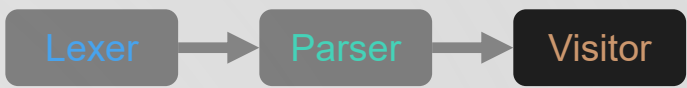
define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
  
```

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}
...	...



Visitor Pattern Example



WRESL Example

```

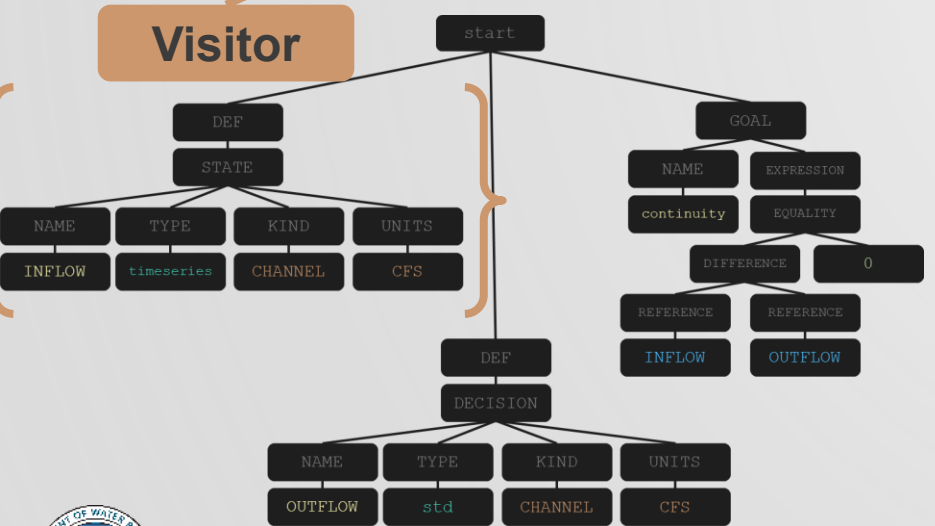
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
  
```

DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}

StateVariable	
String name	INFLOW
Type type	timeseries
String kind	CHANNEL
String units	CFS



Visitor Pattern Example



```
WRESL Example
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

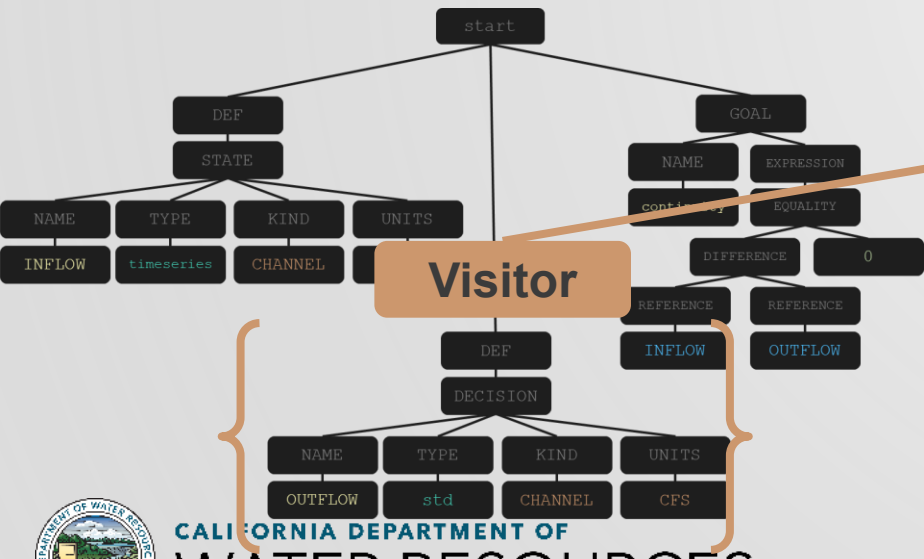
define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
```

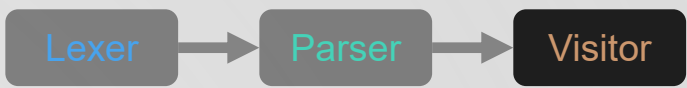
DEF	define
NAME	INFLOW
OPEN_BRACE	{
NEW_LINE	
TIMESERIES	timeseries
NEW_LINE	
KIND	kind
STRING	CHANNEL
NEW_LINE	
UNITS	units
STRING	CFS
NEW_LINE	
CLOSE_BRACE	}

StateVariable	
String name	INFLOW
Type type	timeseries
String kind	CHANNEL
String units	CFS

DecisionVariable	
String name	OUTFLOW
Float lowBound	0
Float upBound	1.0e38
String kind	CHANNEL
String units	CFS



Visitor Pattern Example



```
WRESL Example
define INFLOW {
  timeseries
  kind 'CHANNEL'
  units 'CFS'
}

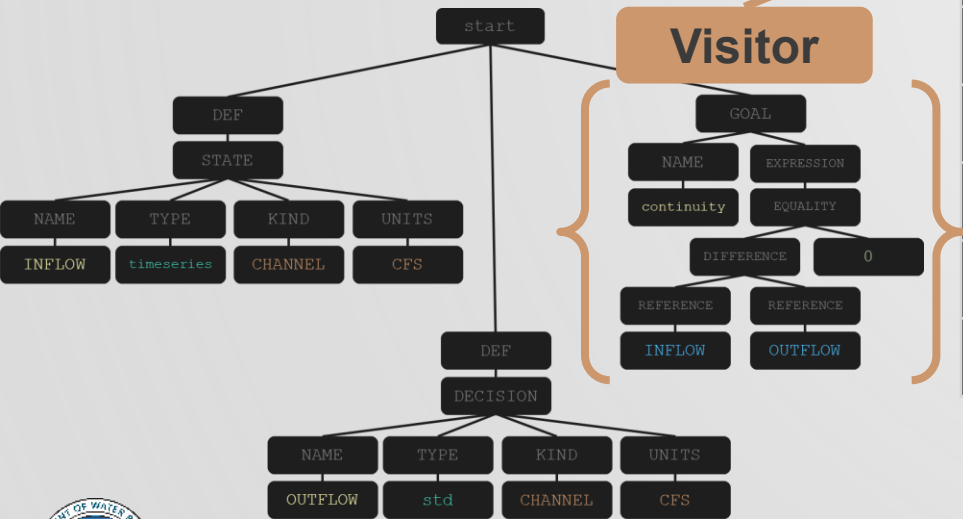
define OUTFLOW {
  std
  kind 'CHANNEL'
  units 'CFS'
}

goal continuity {
  INFLOW - OUTFLOW = 0
}
```

DEF	define
NAME	INFLOW
OPEN BRACE	{
NEW LINE	
TIMESERIES	timeseries
NEW LINE	
KIND	kind
STRING	CHANNEL
NEW LINE	
UNITS	units
STRING	CFS
NEW LINE	
CLOSE BRACE	}

StateVariable	
String name	INFLOW
Type type	timeseries
String kind	CHANNEL
String units	CFS

Goal	
String name	continuity
Expression exp	<pointer>



Visitor

DecisionVariable	
String name	OUTFLOW
Float lowBound	0
Float upBound	1.0e38
String kind	CHANNEL
String units	CFS



Future WRESL+ Development

- Supporting new language features
 - Easier syntax for continuity equations
 - Easier use of external functions/tools
- Improved tooling and diagnostics
 - Python-based parsers to empower model exploration
 - Clearer error messages and static analysis
- Enabling long-term WRIMS evolution
 - Modern tooling
 - Lower maintenance costs



Summary

Modern

Migrating from ANTLR 3 to 4 brings the WRESL+ compiler onto an actively maintained, well-supported foundation

Architectural Improvements

Separation of grammar and lexer/parser improves maintainability

Future Ready

New language features and better tooling are now possible

Better User Experience

Clearer error messages, and modern IDE support are on the way



Follow us on social media



CADWR



CA_DWR



calwater



cadepartmentofwaterresources



CALIFORNIA DEPARTMENT OF
WATER RESOURCES