Ur/Web: A Statically-Typed Language for Building Web Applications from Components

Adam Chlipala
Emerging Languages Camp 2010

Web Applications: A Steaming Pile of Text

HTML, CSS, JavaScript, SQL, URLs, JSON, ...

Strings, strings, strings, ...

Web App Developer

RB-1
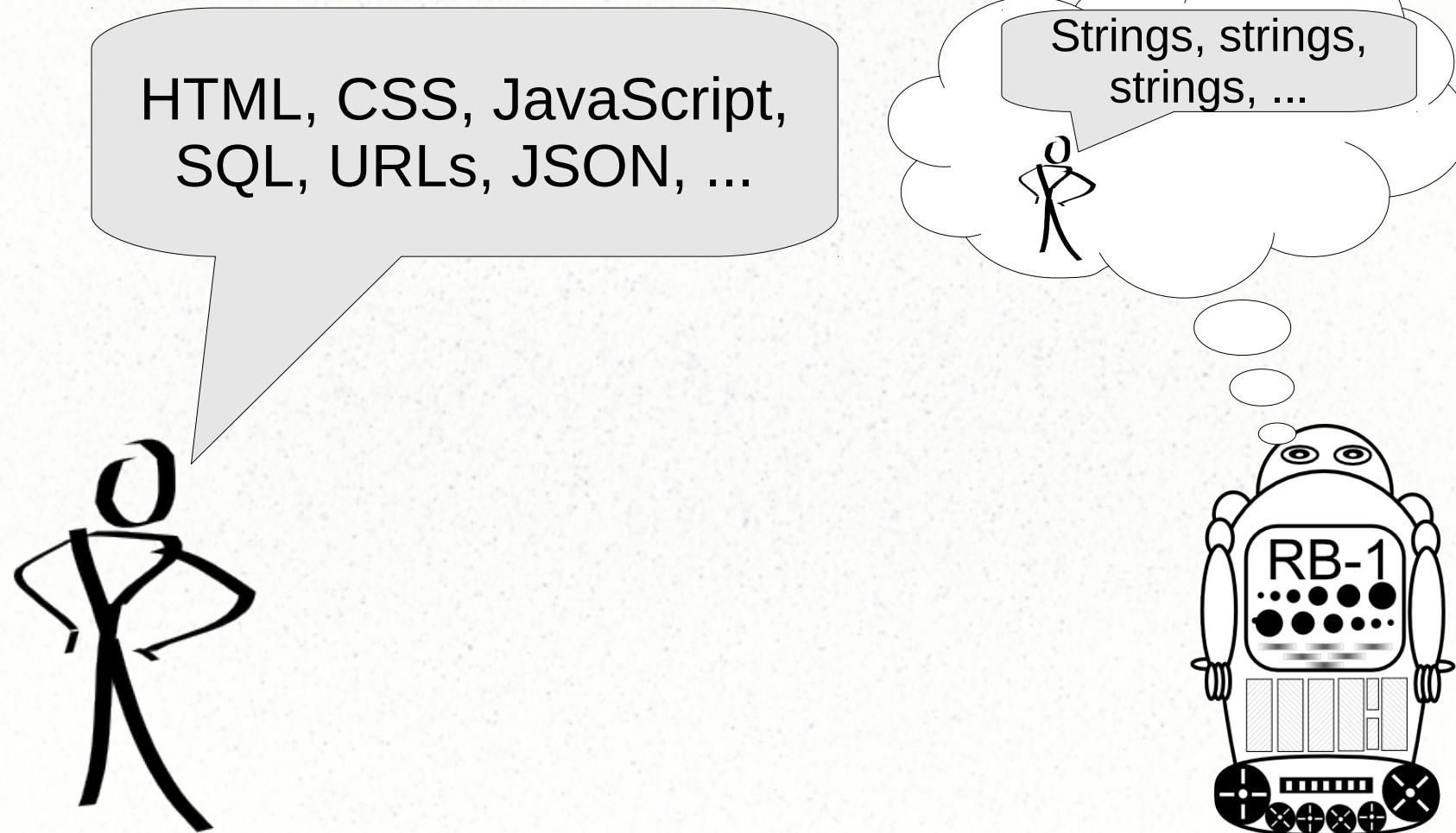
Compiler/Interpreter

Executive Summary

Ur/Web is a **domain-specific language**

with a **fancy static type system**

with **first-class support** for Web app architecture

including **strong encapsulation**

and **statically-checked metaprogramming**

with **security by construction**

Strong Encapsulation

Hash Table   **Module Interface**   **Client Code**

Insert

Lookup
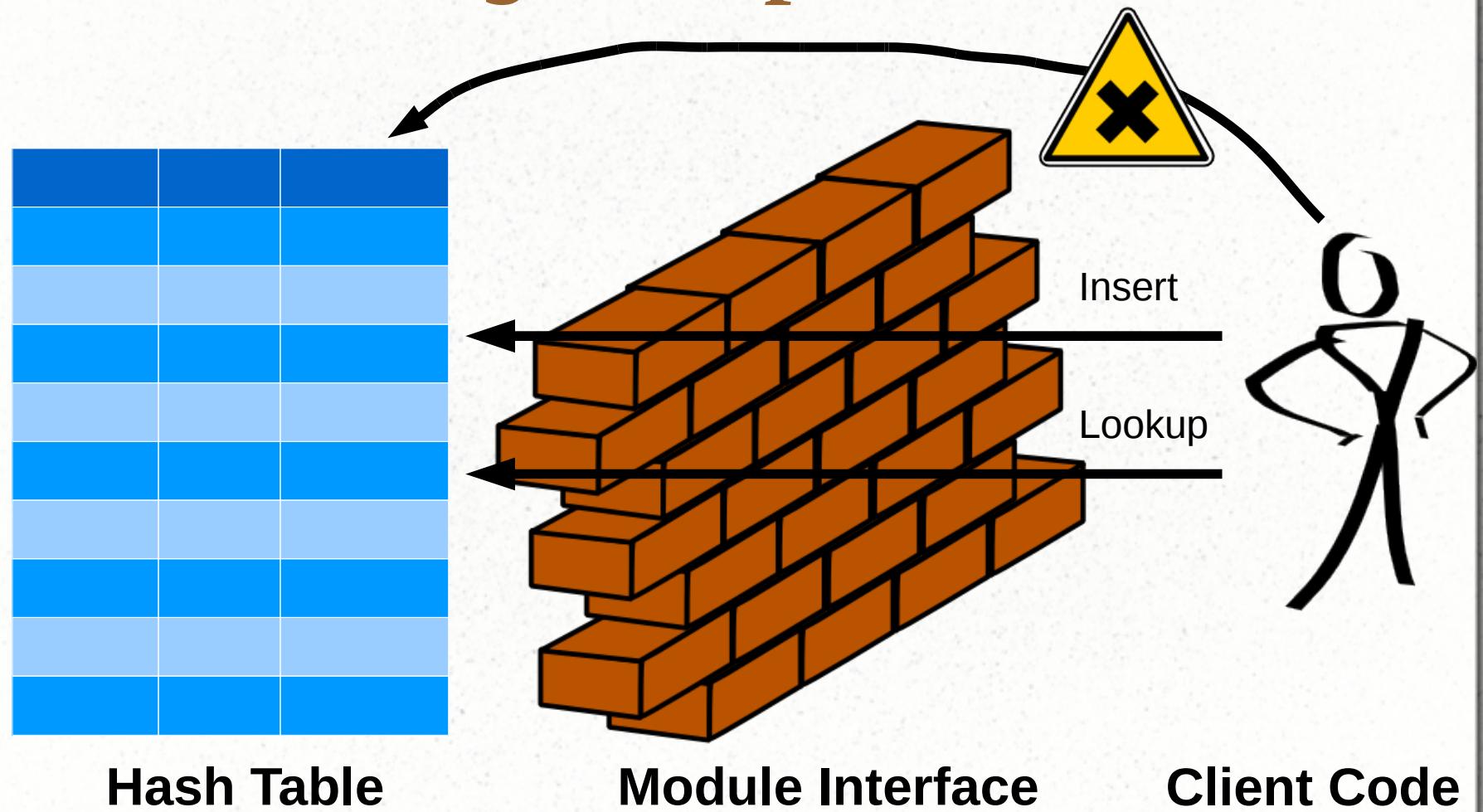
Web 1.0 Encapsulation
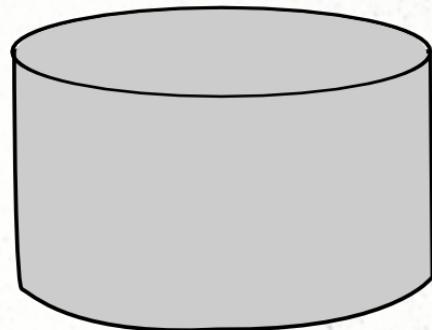
Users Database Table
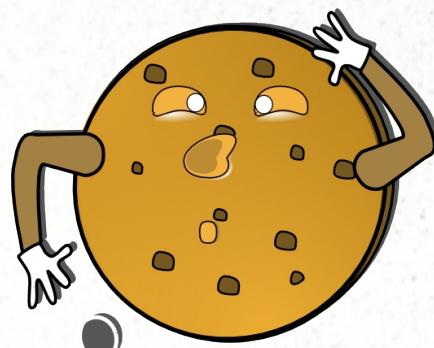
**Login Cookie**
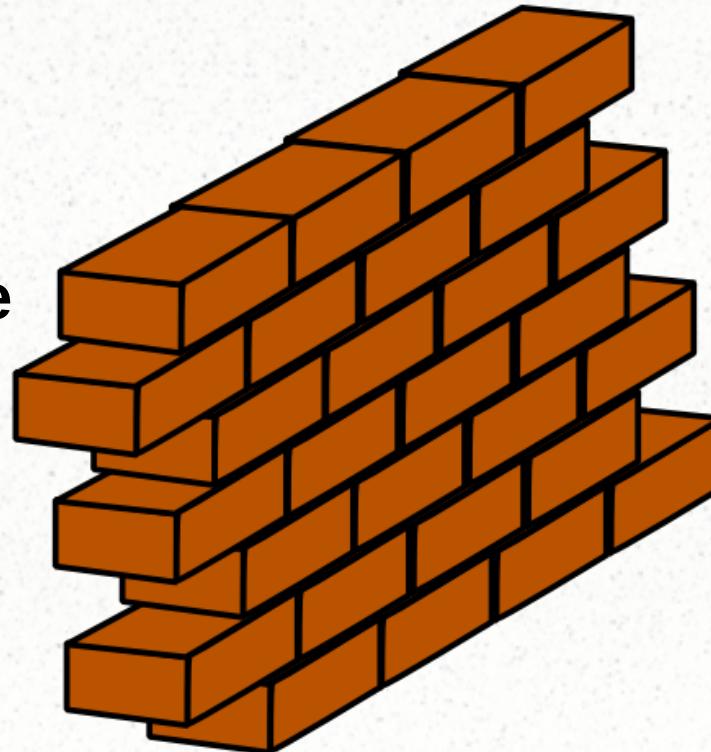
Module Interface

Client Code

Web 2.0 Encapsulation

**Subtree of
Dynamic Page
Structure**
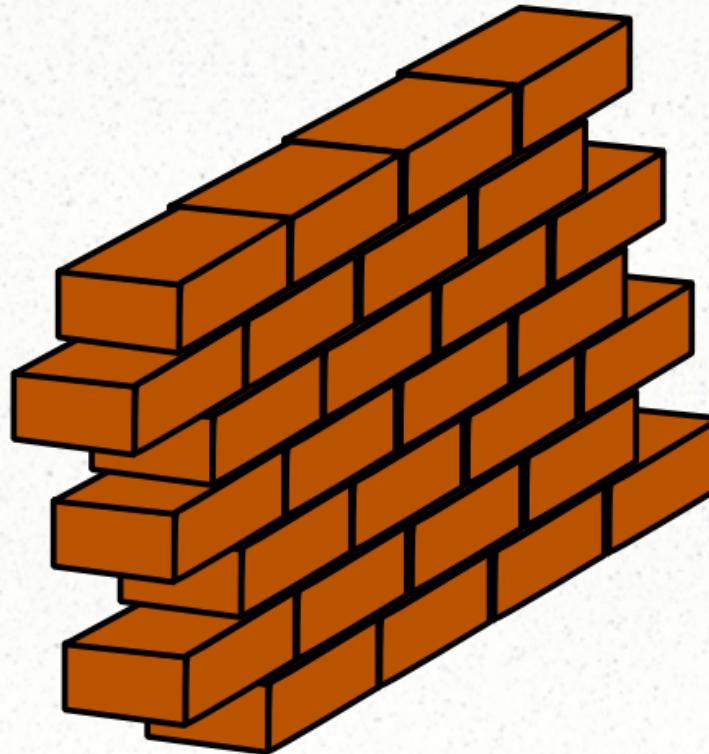
Module Interface

Client Code

Automatic Admin Interface

Id	A	B	C	D

SQL Table Schema

Metaprogram

Crud1

ID	A	B	C	D		
115	2	1	65	True	[Update]	[Delete]
123	10	1000	100	True	[Update]	[Delete]

A: [_____]
B: [_____]
C: [_____]
D: ☐
[Submit Query]

In-Browser Spreadsheet

		No sort	Id	A	B	C	D	E	F	2A	Link
Update	Delete		138	1	4	False	default		NULL	2	Go
Update	Delete		137	0		True	default		NULL	0	Go
Save	Cancel		136	56	qqq	☑	default ⇕		NULL ⇕	…	…
Update	Delete		135	0		False	further		NULL	0	Go
Update	Delete		134	7657		True	default		NULL	15314	Go
Update	Delete		140	0	141	False	other		NULL	0	Go
Update	Delete		157	0		False	default		NULL	0	Go
Aggregates				7715		False					
Filters						⇕	⇕		⇕		

Pages: [1] | **2**

New row   Refresh

Metaprogramming in a Nutshell

Compile-time Programming Language
(AKA, type system)
(not Turing complete, but includes basic lambda calculus)
Crucial feature: **type-level records!**

Included in

Run-time Programming Language
(Turing complete)

Key Property 1:

These types are expressive
enough to guarantee absence of
code injection, abstraction
violation, etc..

Key Property 2:

Effective static checking that the
metaprogram really has the
claimed type

Metadata
controlling
code
generation
options

Metaprogram

Sub-Web

Metadata's
type

Type-level program

Sub-Web's
type

9

An Example Application

Id   A   B   C   D

SQL Table Schema

Crud1

ID	A	B	C	D		
115	2	1	65	True	[Update]	[Delete]
123	10	1000	100	True	[Update]	[Delete]

A: [   ]
B: [   ]
C: [   ]
D: ☐
[Submit Query]

Metaprogram →
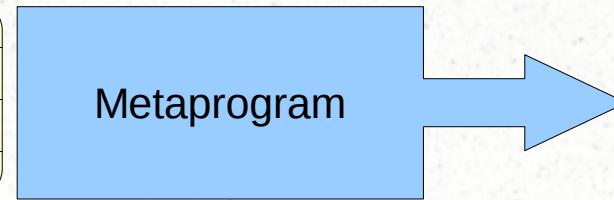
```
table t1 : {Id : int, A : int, B : string,
            C : float, D : bool}
  PRIMARY KEY Id

open Crud.Make(struct
                 val tab = t1

                 val title = "Crud1"

                 val cols = {A = Crud.int "A",
                             B = Crud.string "B",
                             C = Crud.float "C",
                             D = Crud.bool "D"}

             end)
```

10

Nice and Speedy
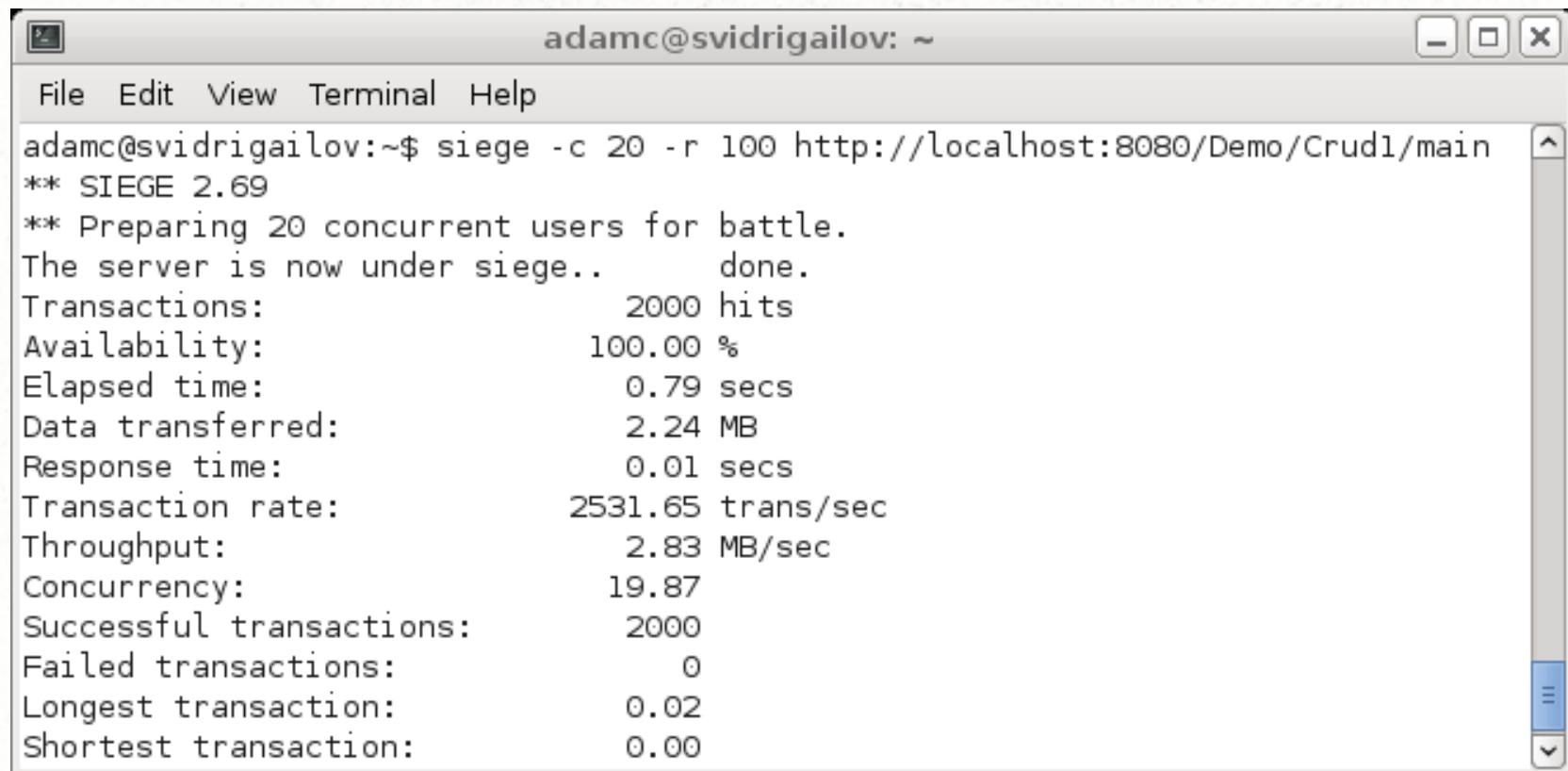
```
adamc@svidrigailov: ~

File  Edit  View  Terminal  Help

adamc@svidrigailov:~$ siege -c 20 -r 100 http://localhost:8080/Demo/Crud1/main
** SIEGE 2.69
** Preparing 20 concurrent users for battle.
The server is now under siege..        done.
Transactions:                       2000 hits
Availability:                     100.00 %
Elapsed time:                       0.79 secs
Data transferred:                   2.24 MB
Response time:                      0.01 secs
Transaction rate:                2531.65 trans/sec
Throughput:                         2.83 MB/sec
Concurrency:                       19.87
Successful transactions:            2000
Failed transactions:                   0
Longest transaction:                0.02
Shortest transaction:               0.00
```

Low Memory Footprint

Ur/Web Available At:

`http://www.impredicative.com/ur/`

Including online demos with syntax-highlighted source code