On Contracts and Sandboxes for JavaScript

Matthias Keil, Peter Thiemann
University of Freiburg, Germany

August 6, 2015
Darmstadt, Germany
89.8 %
of all web sites use JavaScript\(^1\)

\(^1\) according to http://w3techs.com/, status of July 2015
89.8 %
of all web sites use JavaScript\textsuperscript{1}

- Most important client-side language for web sites
- Web-developers rely on third-party libraries
  - e.g. for calendars, maps, social networks

\textsuperscript{1}according to http://w3techs.com/, status of July 2015
NASA finds 'Earth's bigger, older cousin'

By Michael Pearsons, CNN

Updated 11:45 AM ET (12:45 GMT) July 24, 2015 | Video Source: CNN/NASA

NASA finds Earth's bigger, older cousin

KEPLER-452B IS 1,400 LIGHT YEARS FROM EARTH

KEpler-452 system

Story highlights

The planet is the most Earth-like yet found in the habitable zone of a star like ours.

It's about 60% larger than our own planet and "almost certainly has an atmosphere".

(CNN) - NASA said Thursday that its Kepler spacecraft has spotted "Earth's bigger, older cousin": the first nearly Earth-size planet to be found in the habitable zone of a star similar to our own.

Though NASA can't say for sure whether the planet is rocky like ours or has water and air, it's the closest thing to Earth yet discovered by the space agency.
NASA finds 'Earth's bigger, older cousin'

By Michael Peerson, CNN
Updated 1443 GMT (21:43 HKT) July 24, 2015 | Video Source: CNN/NASA

KEPLER-452B IS 1,400 LIGHT YEARS FROM EARTH

Story highlights

The planet is the most Earth-like yet found in the habitable zone of a star like ours.

It's about 60% larger than our own planet and "almost certainly has an atmosphere."

(CNN) — NASA said Thursday that its Kepler spacecraft has spotted "Earth's bigger, older cousin"—the first nearly Earth-size planet to be found in the habitable zone of a star similar to our own.

Though NASA can't say for sure whether the planet is rocky like ours or has water and air, it's the closest we've come to finding another Earthlike planet.
NASA finds 'Earth's bigger, older cousin'

By Michael Peerson, CNN

Updated 1443 GMT (2135 HKT) July 24, 2015 | Video Source: CNN/NASA

KEPLER-452B IS 1,400 LIGHT YEARS FROM EARTH

Story highlights

The planet is the most Earth-like yet found in the habitable zone of a star like ours.

It's about 60% larger than our own planet and "almost certainly has an atmosphere."

(CNN) — NASA said Thursday that its Kepler spacecraft has spotted "Earth's bigger, older cousin": the first nearly Earth-size planet to be found in the habitable zone of a star similar to our own.

Though NASA can't say for sure whether the planet is rocky like ours or has water and air, it's the closest
NASA finds 'Earth's bigger, older cousin'

By Michael Peerson, CNN

Updated 1445 GMT (2135 HKT) July 24, 2015 | Video Source: CNN/NASA

The planet is the most Earth-like yet found in the habitable zone of a star like ours.

It's about 60% larger than our own planet and "almost certainly has an atmosphere".

(CNN) – NASA said Thursday that its Kepler spacecraft has spotted "Earth's bigger, older cousin": the first nearly Earth-size planet to be found in the habitable zone of a star similar to our own.

Though NASA can't say for sure whether the planet is rocky like ours or has water and air, it's the closest
NASA finds 'Earth's bigger, older cousin'

By Michael Peeler, CNN

Updated 14:50 GMT (21:50 HKT) July 24, 2015 | Video Source: CNN/NASA

NASA finds Earth's bigger, older cousin

Mysteriously 'heartbeat' caused by sunspot cycle

Whole new look at Pluto

New Horizons passes Pluto after 3 billion-mile journey

See NASA New Horizons arrives at Pluto

Story highlights

The planet is the most Earth-like yet found in the habitable zone of a star like ours.

It's about 60% larger than our own planet and "almost certainly has an atmosphere."

(CNN) — NASA said Thursday that its Kepler spacecraft has spotted "Earth's bigger, older cousin": the first nearly Earth-size planet to be found in the habitable zone of a star similar to our own.

Though NASA can't say for sure whether the planet is rocky like ours or has water and air, it's the closest
NASA finds 'Earth's bigger, older cousin'

By Michael Peerson, CNN

Updated 1435 GMT (2135 HKT) July 24, 2015 | Video Source: CNN/NASA

KEPLER-452B IS 1,400 LIGHT YEARS FROM EARTH

Story highlights

The planet is the most Earth-like yet found in the habitable zone of a star like ours.

It's about 60% larger than our own planet and "almost certainly has an atmosphere." (CNN) - NASA said Thursday that its Kepler spacecraft has spotted "Earth's bigger, older cousin": the first nearly Earth-size planet to be found in the habitable zone of a star similar to our own.

Though NASA can't say for sure whether the planet is rocky like ours or has water and air, it's the closest match to Earth so far discovered outside our solar system.
JavaScript issues

- Dynamic programming language
  - Code is accumulated by dynamic loading
  - e.g. eval, mashups

Problems

1. Side effects may cause unexpected behavior
2. Program understanding and maintenance is difficult
3. Libraries may get access to sensitive data
4. User code may be prone to injection attacks
JavaScript issues

- Dynamic programming language
  - Code is accumulated by dynamic loading
  - e.g. eval, mashups
- JavaScript has no security awareness
  - No namespace or encapsulation management
  - Global scope for variables/ functions
  - All scripts have the same authority

Problems
1. Side effects may cause unexpected behavior
2. Program understanding and maintenance is difficult
3. Libraries may get access to sensitive data
4. User code may be prone to injection attacks
JavaScript issues

- Dynamic programming language
  - Code is accumulated by dynamic loading
  - e.g. eval, mashups
- JavaScript has no security awareness
  - No namespace or encapsulation management
  - Global scope for variables/ functions
  - All scripts have the same authority

Problems

1. Side effects may cause unexpected behavior
2. Program understanding and maintenance is difficult
3. Libraries may get access to sensitive data
4. User code may be prone to injection attacks
Key challenges of present research

- All-or-nothing choice when including code
- Isolation guarantees noninterference
- Some scripts must have access the application state or are allowed to change it

Goals

1. Manage untrusted JavaScript Code
2. Control the use of data by included scripts
3. Reason about effects of included scripts
Shortcomings

- Static verifiers are imprecise because of JavaScript’s dynamic features or need to restrict JavaScript’s dynamic features.
- Interpreter modifications guarantee full observability but need to be implemented in all existing engines.

- Implemented as a library in JavaScript.
- Library can easily be included in existing projects.
- All aspects are accessible through an API.
- No source code transformation or change in the JavaScript run-time system is required.
Timeline

**2011**
- **JSConTest**
  - Access Permission Contracts for Scripting Languages

**2013**
- **JSConTest2**
  - Efficient Access Analysis Using JavaScript Proxies

**2015**
- **TreatJS**
  - Higher-Order Contracts for JavaScript
  - **TreatJS-Sandbox**
  - Transaction-based Sandboxing of JavaScript

**Ongoing Work**
- **Temporal Contracts, Lemma Contracts, Invariants**
JSConTest
Access Permission Contracts for Scripting Languages
Investigate effects of unfamiliar function
Type and effect contracts with run-time checking
Summarizes observed access traces to a concise description
Effect contracts specifying allowed access paths

Type and effect contracts

```javascript
/*c (obj, obj) -> any with [x.b,y.a] */
function f(x, y) {
  y.a = 1;
  y.b = 2; // violation
}
```
Shortcomings of JSConTest

- Implemented by an offline code transformation
  - Partial interposition (dynamic code, `eval`, `with`, ...)
  - Tied to a particular version of JavaScript
  - Transformation hard to maintain

- Special contract syntax
  - Requires a special JavaScript parser

- Efficiency issues
  - Naive representation of access paths
  - Wastes memory and impedes scalability
Efficient Access Analysis Using JavaScript Proxies
Redesign and reimplementation of JSConTest based on JavaScript proxies

Advantages

- Full interposition for the full language
  - Including dynamically loaded code and `eval`
- Safe for future language extensions
  - No transformation to maintain
- Runs faster in less memory
  - Efficient representation of access paths
  - Incremental path matching
- Maintenance is simplified
  - No custom syntax for contracts
var obj = APC.permit('(a.?+b*)', {a:{b:5},b:{b:11}});

a = obj.a; // APC.permit('?', {b:5});
a.b = 3;

- **APC** encapsulates JSConTest2
- **permit** wraps an object with a permission. Arguments:
  1. Permission encoded in a string
  2. Object that is protected by the permission
- Contract specifies permitted access paths
  - Last property is readable/ writeable
  - Prefix is read-only
  - Not addressed properties are neither readable nor writeable
  - Read-only paths possible (denotes a non-existing property)
Proxy Membrane

Contract: $C$
Path: $\mathcal{P}$

ProxyA

TargetA

Contract: $\partial_x C$
Path: $\mathcal{P}$. $x$

Contract: $\partial_y C$
Path: $\mathcal{P}$. $y$

Contract: $(\partial_z \partial_x C) \& (\partial_y C)$
Path: $\mathcal{P}$. ($x$. $z$ $|$ $y$)
Proxy Membrane

Contract: $C$
Path: $P$

Contract: $\partial_x C$
Path: $P.x$

ProxyA \hspace{2cm} TargetA

ProxyB \hspace{2cm} TargetB

Matthias Keil, Peter Thiemann
On Contracts and Sandboxes
August 6, 2015
Proxy Membrane

Contract: $C$
Path: $\mathcal{P}$

Contract: $\partial_x C$
Path: $\mathcal{P}.x$

Contract: $\partial_y C$
Path: $\mathcal{P}.y$
Proxy Membrane

Contract: $C$
Path: $P$

Contract: $\partial_x C$
Path: $P.x$

Contract: $(\partial_z \partial_x C) \& (\partial_y C)$
Path: $P.(x.z|y)$

Matthias Keil, Peter Thiemann
On Contracts and Sandboxes
August 6, 2015 14 / 43
The JSConTest2 Approach

- Implementation based on the JavaScript Proxy API
- Shortcomings of previous, translation-based implementation avoided
- Full interposition of contracted objects
  - Proxy intercepts all operations
  - Proxy-handler contains a contract and a path set
  - Forwards the operation or signals a violation
- Returned object contains the remaining contract (Membrane)
- Access contracts are regular expressions on literals
  - Each literal defines a property access
  - The language defines a set of permitted access paths
TreatJS

Higher-Order Contracts for JavaScript
Introduction

- Language embedded contract system for JavaScript
- Enforced by run-time monitoring

- Specifies the interface of a software component
- Pre- and postconditions

- Standard abstractions for higher-order-contracts (base, function, and dependent contracts) [Findler, Felleisen’02]
- Systematic blame calculation
- Side-effect free contract execution
- Contract constructors generalize dependent contracts
Base Contracts are built from predicates
Specified by a plain JavaScript function

```javascript
function isNumber (arg) {
    return (typeof arg) === 'number';
};
var _Number_ = Contract.Base(isNumber);

assert(1, _Number_); ✓
assert('a', _Number_); ✗ blame the subject
```

Subject \( v \) gets blamed for Base Contract \( \mathcal{B} \) iff:
\[ \mathcal{B}(v) \neq \text{true} \]
Function Contract [Findler, Felleisen’02]

// Number × Number → Number
function plus (x, y) {
    return (x + y);
}

Matthias Keil, Peter Thiemann
On Contracts and Sandboxes August 6, 2015 19 / 43
// Number × Number → Number

function plus (x, y) {
  return (x + y);
}

var plus = assert(plus, Contract.Function([_Number_, _Number_, _Number_]));
Function Contract [Findler,Felleisen’02]

// Number × Number → Number
function plus (x, y) {
    return (x + y);
}

plus('a', 'a'); X blame the context

- Context gets \textit{blamed} for $C \rightarrow C'$ iff:
  Argument $x$ gets \textit{blamed} for $C$ (as subject)
Function Contract [Findler,Felleisen’02]

```javascript
// Number × Number → Number
function plusBroken (x, y) {
    return (x>0 && y>0) ? (x + y) : 'Error';
}

plusBroken(0, 1); \* blame the subject
```

- Subject $f$ gets blamed for $C \rightarrow C'$ iff:
  $\neg (\text{Context gets blamed } C) \land (f(x) \text{ gets blamed } C')$
New!
Function *plus* works for strings, too

- Requires to model overloading and multiple inheritances

```c
// Number × Number → Number
function plus (x, y) {
  return (x + y);
}

plus('a', 'a'); // blame the context
```
Combinations of Contracts

- No support for arbitrary combination of contracts
- Racket supports \texttt{and/c} and \texttt{or/c}
- Attempt to extend conjunction and disjunction to higher-order contracts
Combinations of Contracts
and/c

- and/c tests any contract
- no value fulfills Number and String at the same time

\( (\text{and/c} (\text{Number} \times \text{Number} \rightarrow \text{Number}) \ (\text{String} \times \text{String} \rightarrow \text{String})) \)

function \( \text{plus} (x, y) \) {
    return \((x + y)\);
}

\( \text{plus}('a', 'a'); \times \text{blame the context} \)
Combinations of Contracts

- `or/c` checks first-order parts and fails unless exactly one (range) contract remains
- Work for disjoint base contracts
- No combination of higher-order contracts
- No support for arbitrary combinations of contracts

```plaintext
(or/c (Number × Number → Number) (String × String → String))
function plus (x, y) {
  return (x + y);
}
```

`plus('a', 'a'); ✓`
Support for arbitrary combination of contracts
  - Combination of base and function contracts
  - Combination of function contracts with a different arity

Intersection and union contracts

Boolean combination of contracts
Intersection Contract

// (Number × Number → Number) ∩ (String × String → String)

function plus (x, y) {
    return (x + y);
}

/\ (Number \times Number \rightarrow Number) \cap (String \times String \rightarrow String)

function plus (x, y) {
    return (x + y);
}

var plus = assert(plus, Contract.Intersection(
    Contract.Function([_Number_, _Number_], _Number_)
    Contract.Function([_String_, _String_], _String_));
Intersection Contract

```javascript
// (Number × Number → Number) ∩ (String × String → String)

function plus(x, y) {
  return (x + y);
}

plus(true, true); ✗ blame the context
```

- Context gets blamed for $C \cap C'$ iff:
  
  (Context gets blamed for $C$) $\land$ (Context gets blamed for $C'$)
// (Number \times Number \rightarrow Number) \cap (String \times String \rightarrow String)

function plusBroken (x, y) {
    return (x \gt 0 && y \gt 0) ? (x + y) : 'Error';
}

plusBroken(0, 1); \times blame the subject

- Subject \ f \ gets \ \underline{blamed} \ for \ C \cap C' \ iff:
  \ (f \ gets \ \underline{blamed} \ for \ C) \ \lor \ (f \ gets \ \underline{blamed} \ for \ C')
A failing contract must not signal a violation immediately.

Violation depends on combinations of failures in different sub-contracts.

```javascript
// (Number → Number) ∩ (String → String)
function addOne (x) {
    return (x + 1);
}

addOne('a');
```
A failing contract must not signal a violation immediately
Violation depends on combinations of failures in different sub-contracts

```
// (Number → Number) ∩ (String → String)
function addOne (x) {
  return (x + 1);
}

addOne('a');
```
A failing contract must not signal a violation immediately.

Violation depends on combinations of failures in different sub-contracts.

```javascript
// (Number → Number) ∩ (String → String)
function addOne (x) {
    return (x + 1);
}

addOne('a'); ✓
```
Contract assertion must connect each contract with the enclosing operations

*Callback* implements a constraint and links each contracts to its next enclosing operation

Reports a record containing two fields, *context* and *subject*

Fields range over $\mathbb{B}_4 = \{\bot, f, t, \top\}$ [Belnap’1977]
Non-Interference

- No syntactic restrictions on predicates
- Problem: Contract may interfere with program execution
- Solution: Predicate evaluation takes place in a sandbox

```javascript
function isNumber (arg) {
  type = (typeof arg);
  return type === 'number';
}

var _Number_ = Contract.Base(isNumber);
```
Non-Interference

- No syntactic restrictions on predicates
- Problem: Contract may interfere with program execution
- Solution: Predicate evaluation takes place in a sandbox

```javascript
function isNumber (arg) {
    type = (typeof arg); ✗ access forbidden
    return type === 'number';
}

var _Number_ = Contract.Base(isNumber);

assert(1, _Number_);
```
All contracts guarantee noninterference

Read-only access is safe

```javascript
var _Array_ = Contract.Base(function (arg) {
    return (arg instanceof Array); // access forbidden
});
```
All contracts guarantee noninterference
Read-only access is safe

```javascript
var _Array_ = Contract.Base(function (arg) {
  return (arg instanceof OutsideArray);
});

var _Array_ = Contract.With({OutsideArray: Array}, _Array_);
```
Contract Constructor

- Building block for dependent, parameterized, abstract, and recursive contracts
- Constructor gets evaluated in a sandbox, like a predicate
- Returns a contract
- No further sandboxing for predicates

```javascript
var __Type__ = Contract.Constructor(function (type) {
    return Contract.Base(function (arg) {
        return (typeof arg) === type;
    });
});

var _Number_ = __Type__('number');
```
TreatJS-Sandbox

Transaction-based Sandboxing of JavaScript
TreatJS-Sandbox

- Language-embedded sandbox for full JavaScript
- Inspired by JSConTest2 and Revocable References
- Adapts SpiderMonkey’s compartment concept to run code in isolation to the application state
- Provides features known from transaction processing in database systems and transactional memory
Sandbox Encapsulation

- A reference is the right to access an object
- Requires to control property read and property write

Sandbox Encapsulation

1. Place a write protection on objects
2. Remove external bindings of functions
Identity Preserving Membrane

ProxyA \rightarrow ProxyB \rightarrow ProxyC \\
\rightarrow TargetA \rightarrow TargetB \rightarrow TargetC
JavaScript Proxies

```javascript
handler.get(target, 'x', proxy);
handler.set(target, 'y', 1, proxy);
handler.get(target, 'y', proxy);

proxy.x;
proxy.y = 1;
proxy.y;

target['x'];
target['y'] = 1;
target['y'];
```
Shadow Objects

```
handler.get(target, 'x', proxy);
handler.set(target, 'y', 1, proxy);
handler.get(target, 'y', proxy);
target['x'];
shadow['y']=1;
shadow['y'];
```
Function decompilation uses the `Function.prototype.toString` method to return a string that contains the source code of that function.

Applying `eval` to the string creates a fresh variant.

A `with` statement places a proxy in top of the scope chain.

The `hasOwnProperty` trap always returns true.
var x = 1;

function f (y) {
  function g () {
    var z = 1;
    return x+y+z;
  }
}

Matthias Keil, Peter Thiemann
var \ x = 1;

\begin{verbatim}
with(sbxglobal){
    function g () {
        var z = 1;
        return x + y + z;
    }
}
\end{verbatim}
Conclusion

- **JSConTest/ JSConTest2**: Effect monitoring for JavaScript
  - Enables to specify effects using access permission contracts
- **TreatJS**: Language embedded, dynamic, higher-order contract system for full JavaScript
  - Support for intersection and union contracts
  - Contract constructors with local scope
- **Sandbox**: Language embedded sandbox for full JavaScript
  - Runs code in a configurable degree of isolation
  - Provides a transactional scope
Ongoing Work

- Temporal/Computation Contracts
- Lemma Contracts
- Invariants
- Different blaming semantics (Lax, Picky, Indy)
Further Challenges

Limitations

- Dynamic contract checking impacts the execution time
- Arbitrary combinations of contracts lead to unprecise error messages

1. Hybrid contract checking
2. Static pre-checking of contracts
3. Optimization, contract rewriting
Appendix
JSConTest
Access Permission Contracts for Scripting Languages
JSConTest2

Efficient Access Analysis Using JavaScript Proxies
Redesign and reimplementation of JSConTest based on JavaScript proxies
Redesign and reimplementation of JSConTest based on JavaScript proxies

Advantages

- Full interposition for the full language
  - Including dynamically loaded code and `eval`
- Safe for future language extensions
  - No transformation to maintain
- Runs faster in less memory
  - Efficient representation of access paths
  - Incremental path path matching
- Maintenance is simplified
  - No custom syntax for contracts
Contracts on Functions

```javascript
var func = APC.permitArgs('arguments.0.(a.?+b*)',
function(arg0) {
    // do something
});
```

- `permitArg` wraps a function with permissions
  1. contract applied to function arguments
  2. function

- Arguments accessed by position `arguments.0`
  - No reliable way to access parameter names
  - Function may use unlisted parameters
  - Parameter names may not be unique
interaction of contracts

\texttt{var obj = APC.permit('((a+a.b)+b.b.@)', \{a:{b:3}, b:{b:5}\});}
\texttt{obj.a = obj.b; // APC.permit('b.@', \{b:5\});}
\texttt{y = obj.a; // APC.permit('b & b.@', \{b:5\});}
\texttt{y.b = 7; \textbf{X violation}}

- Line 2 reads \texttt{obj.b} and writes \texttt{obj.a}
- Afterwards, \texttt{obj.b} and \texttt{obj.a} are aliases
- JSConTest2 enforces \textit{both} contracts reaching \texttt{obj.b} and \texttt{obj.a}
- \texttt{obj.a} carries contract \texttt{!'(e+b)&b.@!} = \texttt{!'b.@!}
- Thus, writing to \texttt{obj.a.b} is not permitted
Limitations

1. Cannot directly protect DOM objects because of the browser’s sandbox

2. Proxies are not transparent with respect to equality, i.e. for distinct proxies == and === returns false, even if the target object is the same
TreatJS

Higher-Order Contracts for JavaScript

Matthias Keil, Peter Thiemann
Object Contract

- Defined by a mapping from property names to contracts
- Represented as a pair of a getter and a setter function
- Getter and Setter apply the property’s contract

\[
\text{var arraySpec = Contract.AObject}({\text{length: Number}});\]
Object Contract

- Defined by a mapping from property names to contracts
- Represented as a pair of a *getter* and a *setter* function
- *Getter* and *Setter* apply the property’s contract

```javascript
var arraySpec = Contract.AObject({length: Number});

var faultyObj = assert({length: '1'}, arraySpec);
```
Object Contract

- Defined by a mapping from property names to contracts
- Represented as a pair of a getter and a setter function
- *Getter* and *Setter* apply the property’s contract

```javascript
var arraySpec = Contract.AObject({length: Number});

var faultyObj = assert({length: '1'}, arraySpec);

var faultyObj.length; X blame the subject
```
Object Contract

- Defined by a mapping from property names to contracts
- Represented as a pair of a *getter* and a *setter* function
- *Getter* and *Setter* apply the property’s contract

```javascript
var arraySpec = Contract.AObject({length: Number});

var faultyObj = assert({length: '1'}, arraySpec);

var faultyObj.length = '2'; x blame the context
```
Function Contract

- *Function Contract* is build from one contract for the domain and one contract for the range of a function.
- An *Object Contract* serves as the domain portion of a *Function Contract* with zero or more arguments.
- *AFunction* defines an *Object Contract* from the array.

```
Contract.AFunction([_Number_, _Number_, _Number_]);
```
Function Contract

- *Function Contract* is built from one contract for the domain and one contract for the range of a function.
- An *Object Contract* serves as the domain portion of a *Function Contract* with zero or more arguments.
- *AFunction* defines an *Object Contract* from the array.

```plaintext
Contract.AFunction([_Number_, _Number_], _Number_);
```

```plaintext
Contract.Function(
  Contract.AObject([_Number_, _Number_]), _Number_));
```
Recursive Contract

- Similar to *Constructors*
- Unrolls the constructor when asserted
- Contract is given as argument to the constructor

```javascript
var __LinkedList__ = Contract.Recursive(
  Contract.Constructor(function(__LinkedList__) {
    return Contract.AObject({
      value:_Number_,
      next:__LinkedList__
    });
  }));
```
// (Number × Number → Number) ∪ (Number × Number → String)

function plusBroken (x, y) {
    return (x > 0 && y > 0) ? (x + y) : 'Error';
}
// (Number × Number → Number) ∪ (Number × Number → String)

function plusBroken (x, y) {
    return (x > 0 && y > 0) ? (x + y) : 'Error';
}

var plusBroken = assert(plusBroken, Contract.Union(
    Contract.Function([_Number_, _Number_], _Number_)
    Contract.Function([_Number_, _Number_], _String_));
// (Number × Number → Number) ∪ (Number × Number → String)

function plusBroken (x, y) {
    return (x>0 && y>0) ? (x + y) : 'Error';
}

plusBroken('a', 'a'); \* blame the context

- Context gets **blamed** for C ∪ C' iff:
  (Context gets **blamed** for C) ∨ (Context gets **blamed** for C')
// (Number × Number → Number) ∪ (Number × Number → String)
function plusBroken (x, y) {
    return (x > 0 && y > 0) ? (x + y) : 'Error';
}

plusBroken(1, 1); ✓
plusBroken(0, 1); ✗ blame the subject
// T × T → T

function plus (x, y) {
    return (x + y);
}

var Plus = assert(plus, Plus);

Plus(Number)(1, 2);
// T × T → T
function plus (x, y) {
    return (x + y);
}

var __Plus__ = Contract.Constructor(function (_Type_) {
    return Contract.Function([Type, Type], Type);
});
// $T \times T \to T$

function plus (x, y) {
    return (x + y);
}

var __Plus__ = Contract.Constructor(function (_Type_) {
    return Contract.Function([Type, Type], Type);
});

var Plus = assert(plus, __Plus__);
// $T \times T \rightarrow T$

```javascript
function plus (x, y) {
    return (x + y);
}

var __Plus__ = Contract.Constructor(function (_Type_) {
    return Contract.Function([_Type_, _Type_], _Type_);
});

var Plus = assert(plus, __Plus__);

Plus(_Number_)(1, 2); ✓
```
Dependent Contract

```javascript
// T × T → T
function plus (x, y) {
  return (x + y);
}

var __Type__ = Contract.Constructor(function (x, y) {
  return Contract.Base(function (arg) {
    return ((typeof x) === (typeof y)) &&
    ((typeof x) === (typeof arg));
  });
});
```
// T × T → T

function plus (x, y) {
    return (x + y);
}

var __Type__ = Contract.Constructor(function(x, y) {
    return Contract.Base(function (arg) {
        return ((typeof x) === (typeof y)) &&
                ((typeof x) === (typeof arg));
    });
});

var plus = assert(plus, Contract.Dependent(__Type__));

plus(1, 2); ✓
// (Number → Number) ∩ (String → String)
function addOneBroken (x) {
    return (x + '1');
}

addOneBroken('a'); ✔
addOneBroken(1); ✗ blame the subject
Callback Graph

\[
\begin{align*}
\land & (\bot, \bot) \\
\bullet & \\
\end{align*}
\]

// (Number \rightarrow Number) \land (String \rightarrow String)
addOneBroken('a');
\[
\text{Callback Graph}
\]

\[
\begin{array}{c}
\text{(\bot,\bot)} \\
\uparrow \\
(\text{\bot,\bot}) \\
\bigcap \\
(\bot,\bot)
\end{array}
\]

// (Number → Number) ∩ (String → String)

addOneBroken('a');
// (Number → Number) ∩ (String → String)
addOneBroken('a');
// (Number → Number) ∩ (String → String)

addOneBroken('a');
Callback Graph

\[ (t, t) \]

```
// (Number → Number) ∩ (String → String)
addOneBroken('a');
```
// (Number → Number) ∩ (String → String)
addOneBroken('a');
Callback Graph

\[ (\bot, \bot) \]

\[ (\bot, \bot) \]

\[ (f, t) \]

\[ (t, f) \]

\[ (t, t) \]

Number Number String

// (Number → Number) ∩ (String → String)
addOneBroken('a');
// (Number → Number) ∩ (String → String)
addOneBroken('a'); ✓
/ (Number → Number) ∩ (String → String)

addOneBroken(1);
\[
\begin{align*}
\text{Callback Graph} & \\
\text{addOneBroken(1);} & \\
(\text{Number} \rightarrow \text{Number}) \cap (\text{String} \rightarrow \text{String})
\end{align*}
\]
Callback Graph

\[(t, t) \quad \cap \quad (\bot, \bot) \quad \cap \quad (f, t) \quad \cap \quad (\bot, \bot) \quad (t, f) \quad (t, t) \]

// (Number → Number) ∩ (String → String)

addOneBroken(1);
Callback Graph

// (Number → Number) ∩ (String → String)
addOneBroken(1); blame the subject
## Scores

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Full</th>
<th>System</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richards</td>
<td>0.391</td>
<td>0.519</td>
<td>11142</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>0.276</td>
<td>0.360</td>
<td>17462</td>
</tr>
<tr>
<td>Crypto</td>
<td>11888</td>
<td>12010</td>
<td>11879</td>
</tr>
<tr>
<td>RayTrace</td>
<td>1.09</td>
<td>1.45</td>
<td>23896</td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>5135</td>
<td>5292</td>
<td>5370</td>
</tr>
<tr>
<td>RegExp</td>
<td>1208</td>
<td>1181</td>
<td>1207</td>
</tr>
<tr>
<td>Splay</td>
<td>20.6</td>
<td>27.8</td>
<td>9555</td>
</tr>
<tr>
<td>SplayLatency</td>
<td>73.1</td>
<td>99.7</td>
<td>6289</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>6234</td>
<td>7159</td>
<td>12612</td>
</tr>
<tr>
<td>pdf.js</td>
<td>9191</td>
<td>9257</td>
<td>9236</td>
</tr>
<tr>
<td>Mandreel</td>
<td>12555</td>
<td>12542</td>
<td>12580</td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>18741</td>
<td>18883</td>
<td>19398</td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>6.80</td>
<td>9.07</td>
<td>23801</td>
</tr>
<tr>
<td>Code loading</td>
<td>6245</td>
<td>6785</td>
<td>9324</td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>3.57</td>
<td>4.67</td>
<td>12528</td>
</tr>
<tr>
<td>zlib</td>
<td>29108</td>
<td>28708</td>
<td>29185</td>
</tr>
<tr>
<td>TypeScript</td>
<td>187</td>
<td>248</td>
<td>11958</td>
</tr>
</tbody>
</table>
## Scores

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Full</th>
<th>w/o C</th>
<th>w/o D</th>
<th>w/o M</th>
<th>w/o P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richards</td>
<td>0.391</td>
<td>0.582</td>
<td>0.782</td>
<td>0.781</td>
<td>0.903</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>0.276</td>
<td>0.409</td>
<td>0.544</td>
<td>0.544</td>
<td>0.625</td>
</tr>
<tr>
<td>Crypto</td>
<td>11888</td>
<td>11912</td>
<td>11914</td>
<td>11986</td>
<td>11979</td>
</tr>
<tr>
<td>RayTrace</td>
<td>1.09</td>
<td>1.82</td>
<td>2.51</td>
<td>2.51</td>
<td>3.02</td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>5135</td>
<td>5126</td>
<td>5205</td>
<td>5233</td>
<td>5242</td>
</tr>
<tr>
<td>RegExp</td>
<td>1208</td>
<td>1205</td>
<td>1199</td>
<td>1212</td>
<td>1178</td>
</tr>
<tr>
<td>Splay</td>
<td>20.6</td>
<td>31.2</td>
<td>42.5</td>
<td>42.5</td>
<td>49.7</td>
</tr>
<tr>
<td>SplayLatency</td>
<td>73.1</td>
<td>109</td>
<td>151</td>
<td>151</td>
<td>177</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>6234</td>
<td>7924</td>
<td>9176</td>
<td>8943</td>
<td>9456</td>
</tr>
<tr>
<td>pdf.js</td>
<td>9191</td>
<td>9548</td>
<td>9156</td>
<td>9222</td>
<td>9152</td>
</tr>
<tr>
<td>Mandreel</td>
<td>12555</td>
<td>12586</td>
<td>12549</td>
<td>12346</td>
<td>12431</td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>18741</td>
<td>18741</td>
<td>18883</td>
<td>19027</td>
<td>18955</td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>6.80</td>
<td>10.8</td>
<td>14.9</td>
<td>14.9</td>
<td>17.7</td>
</tr>
<tr>
<td>Code loading</td>
<td>6245</td>
<td>6937</td>
<td>7372</td>
<td>7335</td>
<td>7533</td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>3.57</td>
<td>5.72</td>
<td>7.80</td>
<td>7.82</td>
<td>9.19</td>
</tr>
<tr>
<td>zlib</td>
<td>29108</td>
<td>29025</td>
<td>29047</td>
<td>28926</td>
<td>29063</td>
</tr>
<tr>
<td>TypeScript</td>
<td>187</td>
<td>290</td>
<td>400</td>
<td>396</td>
<td>463</td>
</tr>
<tr>
<td>Benchmark</td>
<td>A</td>
<td>Contract</td>
<td>I</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>Richards</td>
<td>24</td>
<td>1599377224</td>
<td>935751200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>54</td>
<td>2319477672</td>
<td>1340451212</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crypto</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RayTrace</td>
<td>42</td>
<td>687240082</td>
<td>509234422</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>3944</td>
<td>89022</td>
<td>68172</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RegExp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Splay</td>
<td>10</td>
<td>11620663</td>
<td>7067593</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SplayLatency</td>
<td>10</td>
<td>11620663</td>
<td>7067593</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NavierStokes</td>
<td>51</td>
<td>48334</td>
<td>39109</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pdf.js</td>
<td>3</td>
<td>15</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mandreel</td>
<td>7</td>
<td>57</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>7</td>
<td>57</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>3206</td>
<td>141669753</td>
<td>97487985</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code loading</td>
<td>5600</td>
<td>34800</td>
<td>18400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>20075</td>
<td>172755100</td>
<td>112664947</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zlib</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TypeScript</td>
<td>4</td>
<td>12673644</td>
<td>8449090</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Benchmark</td>
<td>Sandbox</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Richards</td>
<td>4678756000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>6702256060</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crypto</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RayTrace</td>
<td>2546172110</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>340860</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RegExp</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Splay</td>
<td>35337965</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SplayLatency</td>
<td>35337965</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NavierStokes</td>
<td>195545</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pdf.js</td>
<td>45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mandreel</td>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>487439925</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code loading</td>
<td>92000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>563324735</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zlib</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TypeScript</td>
<td>42245450</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Matthias Keil, Peter Thiemann
On Contracts and Sandboxes
August 6, 2015
<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Callback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richards</td>
<td>3351504000</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>4744203248</td>
</tr>
<tr>
<td>Crypto</td>
<td>13</td>
</tr>
<tr>
<td>RayTrace</td>
<td>2190186074</td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>309120</td>
</tr>
<tr>
<td>RegExp</td>
<td>0</td>
</tr>
<tr>
<td>Splay</td>
<td>26231845</td>
</tr>
<tr>
<td>SplayLatency</td>
<td>26231845</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>177197</td>
</tr>
<tr>
<td>pdf.js</td>
<td>39</td>
</tr>
<tr>
<td>Mandreel</td>
<td>128</td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>128</td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>399084085</td>
</tr>
<tr>
<td>Code loading</td>
<td>70400</td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>469141435</td>
</tr>
<tr>
<td>zlib</td>
<td>0</td>
</tr>
<tr>
<td>TypeScript</td>
<td>33796350</td>
</tr>
</tbody>
</table>
## Timings

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Full</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richards</td>
<td>1 day, 18 hours, 21 min, 20 sec</td>
<td>8 sec</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>2 days, 10 hours, 36 min, 49 sec</td>
<td>4 sec</td>
</tr>
<tr>
<td>Crypto</td>
<td>9 sec</td>
<td>8 sec</td>
</tr>
<tr>
<td>RayTrace</td>
<td>23 hours, 12 min, 37 sec</td>
<td>4 sec</td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>1 min, 9 sec</td>
<td>53 sec</td>
</tr>
<tr>
<td>RegExp</td>
<td>9 sec</td>
<td>8 sec</td>
</tr>
<tr>
<td>Splay</td>
<td>19 min, 19 sec</td>
<td>3 sec</td>
</tr>
<tr>
<td>SplayLatency</td>
<td>19 min, 19 sec</td>
<td>3 sec</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>11 sec</td>
<td>4 sec</td>
</tr>
<tr>
<td>pdf.js</td>
<td>6 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>Mandreel</td>
<td>5 sec</td>
<td>5 sec</td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>5 sec</td>
<td>5 sec</td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>4 hours, 28 min, 28 sec</td>
<td>5 sec</td>
</tr>
<tr>
<td>Code loading</td>
<td>12 sec</td>
<td>9 sec</td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>5 hours, 19 min, 49 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>zlib</td>
<td>11 sec</td>
<td>11 sec</td>
</tr>
<tr>
<td>TypeScript</td>
<td>22 min, 46 sec</td>
<td>24 sec</td>
</tr>
<tr>
<td>Benchmark</td>
<td>Predicate</td>
<td>Baseline</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Richards</td>
<td>1 day, 6 hours, 59 min, 42 sec</td>
<td>8 sec</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>1 day, 20 hours, 58 min, 17 sec</td>
<td>4 sec</td>
</tr>
<tr>
<td>Crypto</td>
<td>8 sec</td>
<td>8 sec</td>
</tr>
<tr>
<td>RayTrace</td>
<td>17 hours, 1 min, 25 sec</td>
<td>4 sec</td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>1 min, 3 sec</td>
<td>53 sec</td>
</tr>
<tr>
<td>RegExp</td>
<td>8 sec</td>
<td>8 sec</td>
</tr>
<tr>
<td>Splay</td>
<td>13 min, 55 sec</td>
<td>3 sec</td>
</tr>
<tr>
<td>SplayLatency</td>
<td>13 min, 55 sec</td>
<td>3 sec</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>9 sec</td>
<td>4 sec</td>
</tr>
<tr>
<td>pdf.js</td>
<td>6 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>Mandreel</td>
<td>5 sec</td>
<td>5 sec</td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>5 sec</td>
<td>5 sec</td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>3 hours, 13 min, 13 sec</td>
<td>5 sec</td>
</tr>
<tr>
<td>Code loading</td>
<td>12 sec</td>
<td>9 sec</td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>3 hours, 52 min, 34 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>zlib</td>
<td>12 sec</td>
<td>11 sec</td>
</tr>
<tr>
<td>TypeScript</td>
<td>17 min, 8 sec</td>
<td>24 sec</td>
</tr>
</tbody>
</table>
TreatJS-Sandbox

Transaction-based Sandboxing of JavaScript
NASA finds 'Earth's bigger, older cousin'

By Michael Pearson, CNN

Updated 14:35 GMT (21:35 HKT) July 24, 2015 | Video Source: CNN/NASA

Story highlights

The planet is the most Earth-like yet found in the habitable zone of a star like ours.

It's about 60% larger than our own planet and "almost certainly has an atmosphere."

(CNN) — NASA said Thursday that its Kepler spacecraft has spotted "Earth's bigger, older cousin": the first nearly Earth-size planet to be found in the habitable zone of a star similar to our own.

Though NASA can't say for sure whether the planet is rocky like ours or has water and air, it's the closest

Motivation

Matthias Keil, Peter Thiemann

On Contracts and Sandboxes

August 6, 2015
Non-interfering Contract Execution

```javascript
var _Number_ = Contract.Base(function (arg) {
    type = (typeof arg);
    return type === 'number';
});
```
Non-interfering Contract Execution

```javascript
var _Number_ = Contract.Base((function (arg) {
    type = (typeof arg);
    return type === 'number';
}));
```

Read-only access

```javascript
var _Array_ = Contract.Base((function (arg) {
    return (arg instanceof OutsideArray);
}));

var _Array_ = Contract.With([{OutsideArray:Array}], _Array_);
```
Application Scenarios
TreatJS Online

TreatJS: Higher-Order Contracts for JavaScript
Online Version

Write your JavaScript programs:

```javascript
function f(x, y) {
    return x + y;
}

var g = CONTRACTSPEC(f,
    function(x) {
        return typeof x === 'number';
    },
    function(x, y) {
        return typeof x === 'number' && typeof y === 'number';
    });

return g(3, '4');
```

Output:

Contract Violation: {} Go(typeOfNumber) 1|1(typeOfNumber)|->3(typeOfNumber)
Please fix:
```javascript
Define:
Callers: (false)
$Callers: (false)
```
function Node (value, left, right) {
    this.value = value;
    this.left = left;
    this.right = right;
}

function heightOf (node) {
    return node ? Math.max(heightOf(node.left)+1), heightOf(node.right)+1) : -1;
}

function setValue (node) {
    node.value=heightOf(node);
    if(node.left) setValue(node.left);
    if(node.right) setValue(node.right);
}
Function Invocation

```javascript
var root = new Node(0, new Node(0), new Node(0));
```

Matthias Keil, Peter Thiemann  
On Contracts and Sandboxes  
August 6, 2015  24 / 43
Function Invocation

```javascript
var root = new Node(0, new Node(0), new Node(0));

var sbx = new Sandbox(this, {/* some parameters */});
```
```javascript
var root = new Node(0, new Node(0), new Node(0));

var sbx = new Sandbox(this, { /* some parameters */ });

sbx.call(heightOf, this, root);
```
function heightOf (node) {
    return node ? Math.max(heightOf(node.left)+1), heightOf(node.right)+1) : −1;
}
function heightOf (node) {
    return node ? Math.max(heightOf(node.left)+1, heightOf(node.right)+1) : -1;
}
function heightOf (node) {
    return node ? Math.max(heightOf(node.left)+1), heightOf(node.right)+1) : -1;
}
function setValue (node) {
    node.value = heightOf(node);
    if (node.left) setValue(node.left);
    if (node.right) setValue(node.right);
}
function setValue (node) {
  node.value = heightOf(node);
  if(node.left) setValue(node.left);
  if(node.right) setValue(node.right);
}
var sbx2 = new Sandbox(this, { /* some parameters */ });

function appendRight (node) {
    node.right = Node('a', Node('b'), Node('c'));
}

sbx2.call(appendRight, this, root);
```
sbx.effectsOf(this).forEach(function(i, e) {print(e)});
```

;;; Effects of this
(1425301383541) has [name=heightOf]
(1425301383541) get [name=heightOf]
(1425301383543) has [name=Math]
(1425301383543) get [name=Math]

sbx.writeeffectsOf(root).foreach(function(i, e) {print(e)});

;;; Write Effects of root
(1425301634992) set [name=value]


```
sbx.hasChanges; // returns true
sbx.changes.foreach(function(i, e) {print(e)});

;;;; All Changes
Change: (1425300876577) set [name=value]@SBX001
```
Inspection a Sandbox

Differences

```
; sbx.hasDifferences; // returns true
sbx.differences.foreach(function(i, e) {print(e)});

;;; Differences of root
Difference: (1425300932388) set [name=value]@SBX001
```
sBX.hasDifferences; // returns true
sBX.differences.foreach(function(i, e) {print(e)});

;;; Differences of root
Difference: (1425300932388) set [name=value]@SBX001
sbx.inConflictWith(sbx2); // returns true
sbx.conflictsWith(sbx2).forEach(function(i, e) {print(e)});

Confict: (1425303937853) get [name=right]@SBX001 –
(1425303937855) set [name=right]@SBX002
Transaction Processing

Commit

```
sbx.commit();
```

Matthias Keil, Peter Thiemann  
On Contracts and Sandboxes  
August 6, 2015  
33 / 43
sbx.rollback();
sbx.revert();
Further Features

- Fine-grained handling of commit, rollback and revert
- Transparent sandboxing
- Snapshot Mode
- Wrapping of values
var node = new Node(/*some sub-nodes */);
var sbx = new Sandbox(this, /*some parameters */);

with(sbxglobal){

    function setValue (node) {
        node.value=heightOf(node);
        setValue(node.left);
        setValue(node.right);
    }

}
A new sandbox starts with a blank environment

- Read access by binding values when instantiating a new sandbox
- Fine-grained write access by committing effects
- Combination with 
  *Access Permission Contracts* and
  *Revocable References* possible
Features

- Enables scripts to run in a configurable degree of isolation
- Enables fine-grained access control
- Provides a transactional scope that performs effect logging
- Effects can be committed to the application state or rolled back
Limitations

- Function recompilation needs a string that contains the source code of that function.
- **Function.prototype.toString** did not work for all functions:
  - A native function did not have a string representation.
  - The **Function.prototype.bind** method creates a new bound function without a string representation.
## Experimental Evaluation

### Scores

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Sandbox</th>
<th></th>
<th>Transparent</th>
<th></th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>w/o E</td>
<td>F</td>
<td>w/o E</td>
<td></td>
</tr>
<tr>
<td>Richards</td>
<td>3919</td>
<td>4270</td>
<td>4070</td>
<td>4078</td>
<td>6618</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>177</td>
<td>333</td>
<td>225</td>
<td>333</td>
<td>4778</td>
</tr>
<tr>
<td>Crypto</td>
<td>1508</td>
<td>1917</td>
<td>1666</td>
<td>1959</td>
<td>11012</td>
</tr>
<tr>
<td>RayTrace</td>
<td>158</td>
<td>211</td>
<td>157</td>
<td>220</td>
<td>3941</td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>624</td>
<td>677</td>
<td>618</td>
<td>663</td>
<td>2648</td>
</tr>
<tr>
<td>RegExp</td>
<td>883</td>
<td>900</td>
<td>887</td>
<td>903</td>
<td>1192</td>
</tr>
<tr>
<td>Splay</td>
<td>697</td>
<td>774</td>
<td>706</td>
<td>1394</td>
<td>1439</td>
</tr>
<tr>
<td>SplayLatency</td>
<td>1986</td>
<td>2567</td>
<td>1983</td>
<td>2276</td>
<td>1234</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>915</td>
<td>865</td>
<td>938</td>
<td>934</td>
<td>11835</td>
</tr>
<tr>
<td>pdf.js</td>
<td>84</td>
<td>121</td>
<td>85</td>
<td>127</td>
<td>9229</td>
</tr>
<tr>
<td>Mandreel</td>
<td>95</td>
<td>122</td>
<td>96</td>
<td>124</td>
<td>8521</td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>455</td>
<td>621</td>
<td>437</td>
<td>632</td>
<td>16562</td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>4413</td>
<td>5081</td>
<td>4577</td>
<td>5027</td>
<td>22307</td>
</tr>
<tr>
<td>Code loading</td>
<td>5123</td>
<td>5264</td>
<td>5148</td>
<td>5202</td>
<td>6721</td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>84</td>
<td>121</td>
<td>82</td>
<td>120</td>
<td>13853</td>
</tr>
<tr>
<td>zlib</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>28970</td>
</tr>
<tr>
<td>TypeScript</td>
<td>3006</td>
<td>3339</td>
<td>2940</td>
<td>3242</td>
<td>11813</td>
</tr>
</tbody>
</table>
## Experimental Evaluation

### Timings

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Sandbox</th>
<th>Transparent</th>
<th>Baseline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Richards</td>
<td>14 sec</td>
<td>16 sec</td>
<td>9 sec</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>263 sec</td>
<td>265 sec</td>
<td>13 sec</td>
</tr>
<tr>
<td>Crypto</td>
<td>59 sec</td>
<td>58 sec</td>
<td>9 sec</td>
</tr>
<tr>
<td>RayTrace</td>
<td>865 sec</td>
<td>854 sec</td>
<td>23 sec</td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>265 sec</td>
<td>249 sec</td>
<td>75 sec</td>
</tr>
<tr>
<td>RegExp</td>
<td>14 sec</td>
<td>16 sec</td>
<td>8 sec</td>
</tr>
<tr>
<td>Splay</td>
<td>32 sec</td>
<td>43 sec</td>
<td>17 sec</td>
</tr>
<tr>
<td>SplayLatency</td>
<td>32 sec</td>
<td>43 sec</td>
<td>17 sec</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>59 sec</td>
<td>61 sec</td>
<td>5 sec</td>
</tr>
<tr>
<td>pdf.js</td>
<td>944 sec</td>
<td>872 sec</td>
<td>9 sec</td>
</tr>
<tr>
<td>Mandreel</td>
<td>683 sec</td>
<td>683 sec</td>
<td>9 sec</td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>683 sec</td>
<td>683 sec</td>
<td>9 sec</td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>30 sec</td>
<td>27 sec</td>
<td>7 sec</td>
</tr>
<tr>
<td>Code loading</td>
<td>14 sec</td>
<td>14 sec</td>
<td>10 sec</td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>1156 sec</td>
<td>1140 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>zlib</td>
<td>-</td>
<td>-</td>
<td>11 sec</td>
</tr>
<tr>
<td>TypeScript</td>
<td>86 sec</td>
<td>91 sec</td>
<td>24 sec</td>
</tr>
<tr>
<td>Benchmark</td>
<td>Wrap</td>
<td>Cache</td>
<td>Effect</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------</td>
<td>-------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Miss</td>
<td>Hit</td>
</tr>
<tr>
<td>Richards</td>
<td>164021</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>DeltaBlue</td>
<td>7348024</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Crypto</td>
<td>2329634</td>
<td>3</td>
<td>1046550</td>
</tr>
<tr>
<td>RayTrace</td>
<td>25519224</td>
<td>3</td>
<td>12757207</td>
</tr>
<tr>
<td>EarleyBoyer</td>
<td>5465</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>RegExp</td>
<td>115474</td>
<td>3</td>
<td>65923</td>
</tr>
<tr>
<td>Splay</td>
<td>466840</td>
<td>3</td>
<td>232019</td>
</tr>
<tr>
<td>SplayLatency</td>
<td>466840</td>
<td>3</td>
<td>232019</td>
</tr>
<tr>
<td>NavierStokes</td>
<td>1839</td>
<td>3</td>
<td>727</td>
</tr>
<tr>
<td>pdf.js</td>
<td>18222921</td>
<td>4</td>
<td>37302</td>
</tr>
<tr>
<td>Mandreel</td>
<td>19954116</td>
<td>3</td>
<td>9974959</td>
</tr>
<tr>
<td>MandreelLatency</td>
<td>19954116</td>
<td>3</td>
<td>9974959</td>
</tr>
<tr>
<td>Gameboy Emulator</td>
<td>467585</td>
<td>4</td>
<td>121250</td>
</tr>
<tr>
<td>Code loading</td>
<td>23829</td>
<td>5</td>
<td>4208</td>
</tr>
<tr>
<td>Box2DWeb</td>
<td>24876419</td>
<td>3</td>
<td>7800731</td>
</tr>
<tr>
<td>zlib</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TypeScript</td>
<td>1114199</td>
<td>3</td>
<td>17</td>
</tr>
</tbody>
</table>