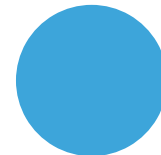
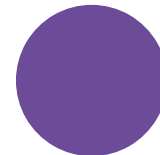
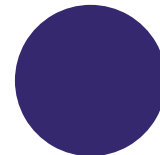
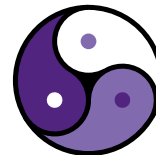


Fuzion

A Language Designed for Safety-Critical Embedded Systems

Dr. Fridtjof Siebert
Tokiwa Software GmbH

Embedded World Conference 2026, 10 March 2026, Nürnberg



Who is this guy?

Fridtjof Siebert



Email: siebert@tokiwa.software
github: [fridis](https://github.com/fridis)
bluesky etc. [@fridis.bsky.social](https://bsky.app/profile/@fridis.bsky.social)
[@fridi@mastodon.social](https://mstdn.social/@fridi)
[fridi_si@instagram](https://www.instagram.com/fridi_si/)

'90-'94	AmigaOberon, AMOK PD
'97	FEC Eiffel Sparc / Solaris
'98-'99	OSF: TurboJ Java Compiler
'00-'01	PhD on real-time GC
'02-'19	JamaicaVM real-time JavaVM VeriFlux static analysis tool
'20-...	Fuzion / Tokiwa Software



Overview

- Motivation
- Fuzion quick intro
- Design-by-Contract
- Effects in Fuzion
- Toolchain and Static Analysis
- Safety-Critical Aspects
- Status



Overview

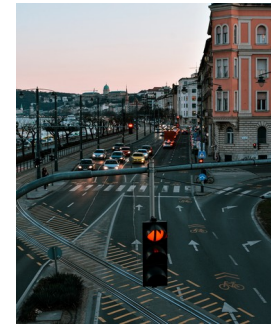
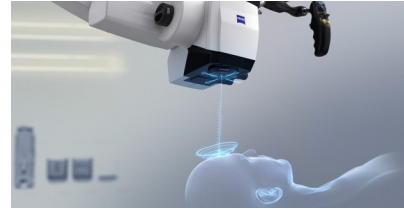
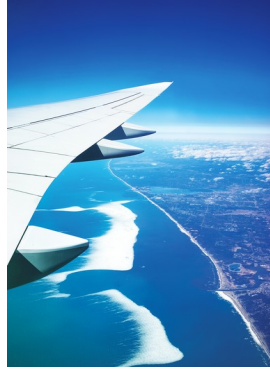
- **Motivation**
- Fuzion quick intro
- Design-by-Contract
- Effects in Fuzion
- Toolchain and Static Analysis
- Safety-Critical Aspects
- Status



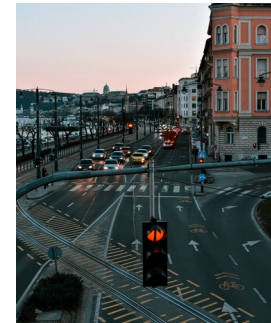
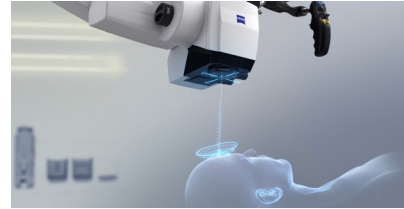
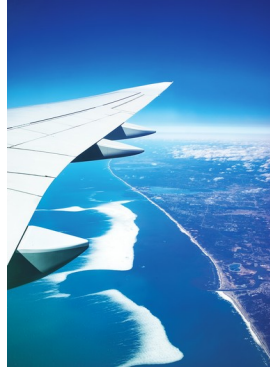
clipart by pascallapalme @ openclipart.org



Motivation: Safety



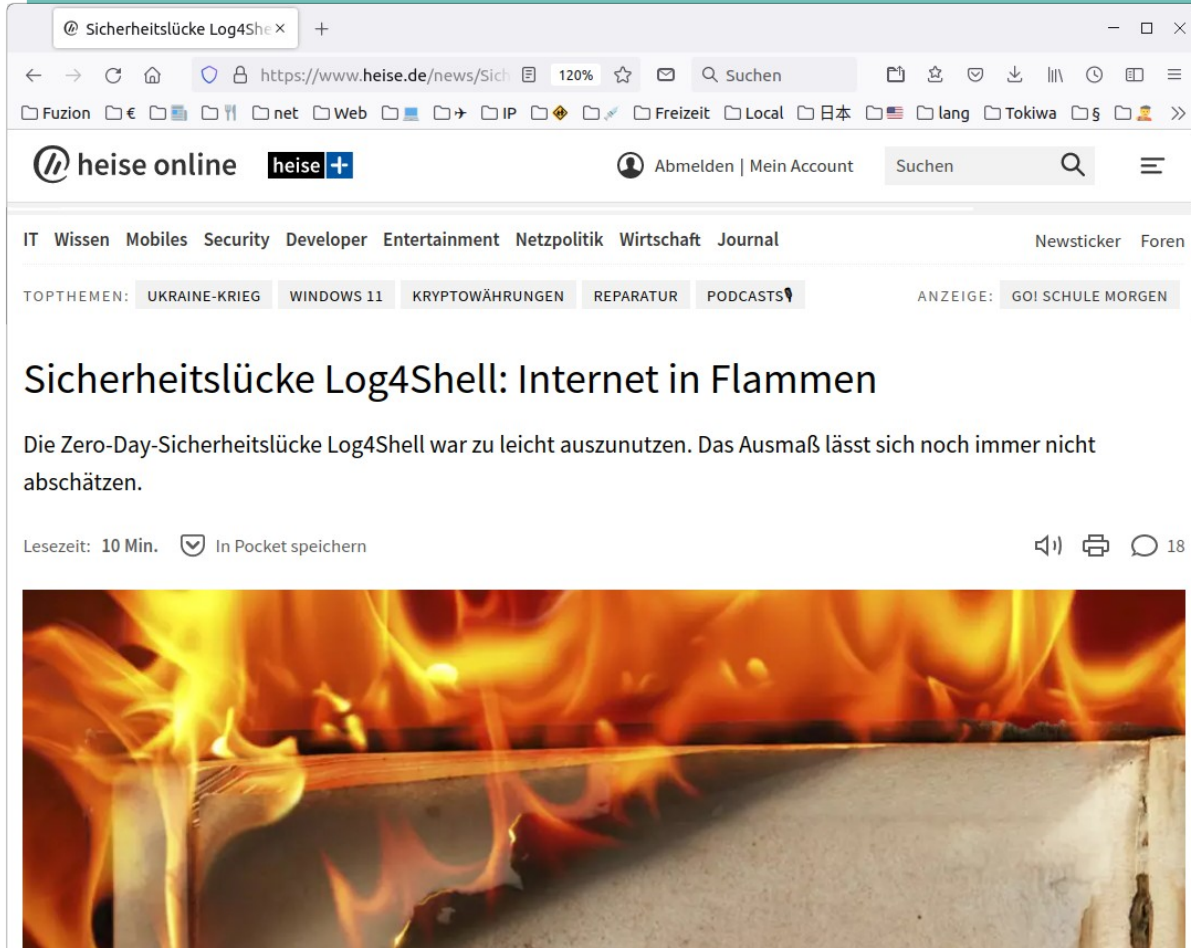
Motivation: Safety



→ we need tools to help certification







Motivation: Security




The screenshot shows a web browser window with the URL <https://www.heise.de/news/Sicherheitslücke-Log4Shell>. The page is from heise online and features a navigation menu with categories like IT, Wissen, Mobiles, Security, Developer, Entertainment, Netzpolitik, Wirtschaft, and Journal. The main headline is "Sicherheitslücke Log4Shell: Internet in Flammen". Below the headline, the text reads: "Die Zero-Day-Sicherheitslücke Log4Shell war zu leicht auszunutzen. Das Ausmaß lässt sich noch immer nicht abschätzen." The article is dated "Lesezeit: 10 Min." and has 18 comments. At the bottom of the article, there is a large image of a fire.

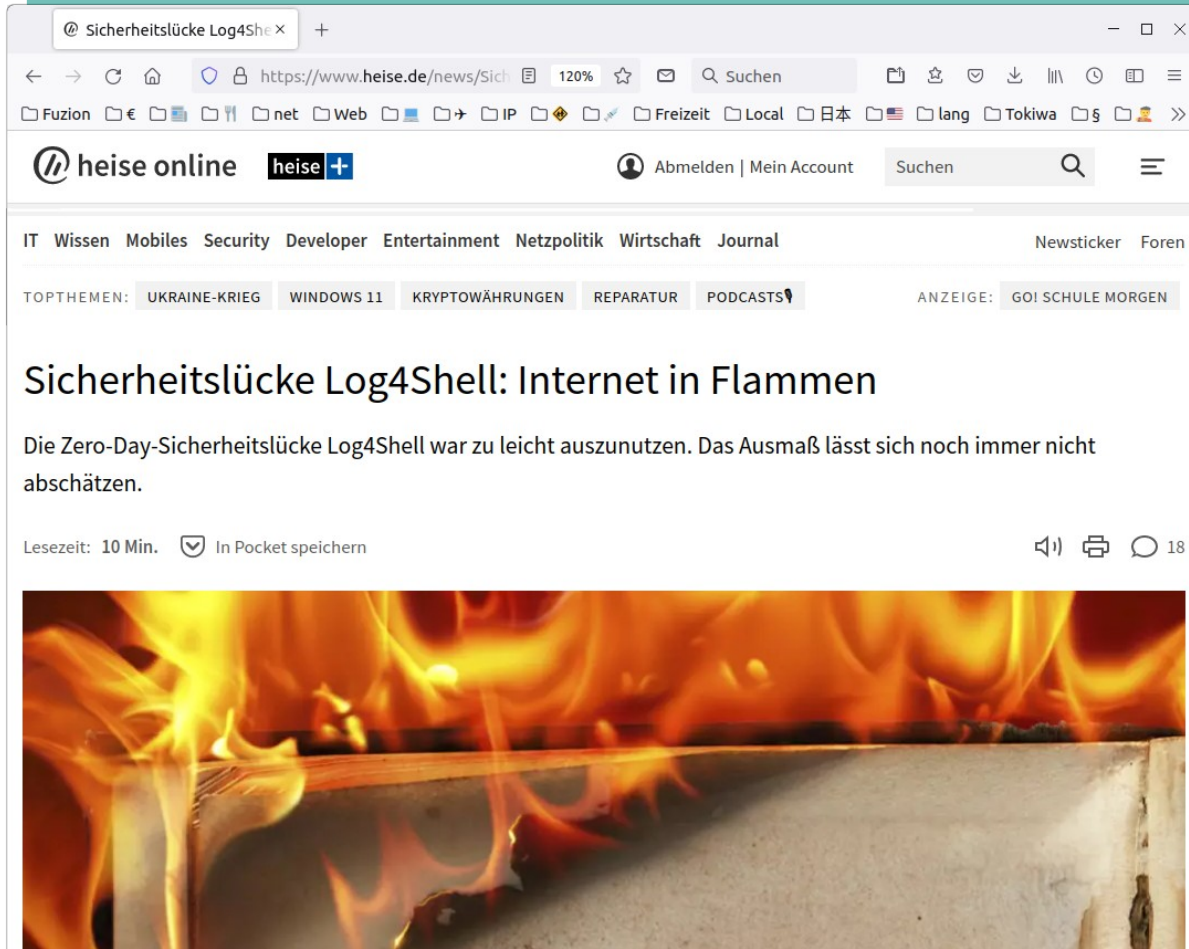
Sicherheitslücke Log4Shell: Internet in Flammen

Die Zero-Day-Sicherheitslücke Log4Shell war zu leicht auszunutzen. Das Ausmaß lässt sich noch immer nicht abschätzen.

Lesezeit: 10 Min.  In Pocket speichern    18



Motivation: Security

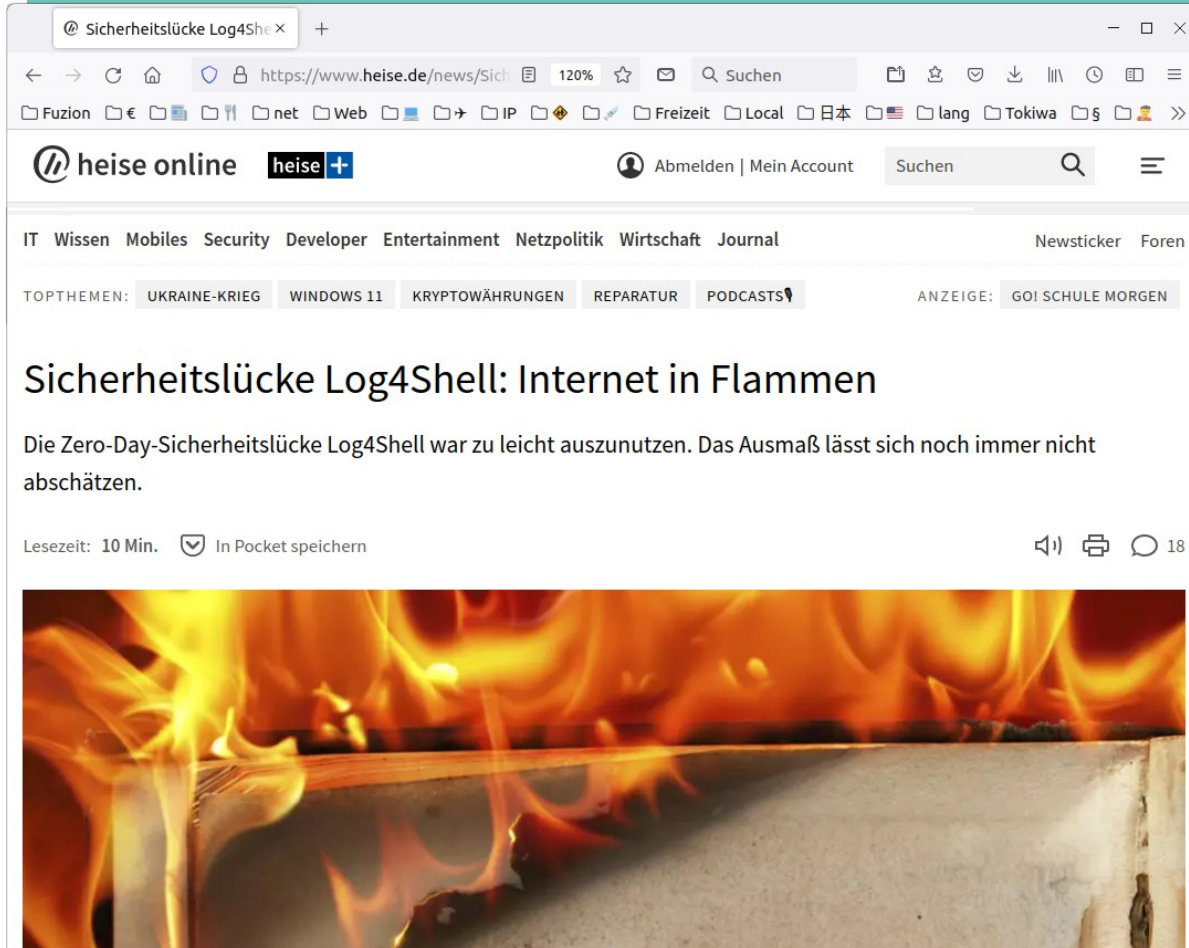


The screenshot shows a web browser window with the address bar displaying "https://www.heise.de/news/Sich...". The page title is "Sicherheitslücke Log4Shell: Internet in Flammen". The article text reads: "Die Zero-Day-Sicherheitslücke Log4Shell war zu leicht auszunutzen. Das Ausmaß lässt sich noch immer nicht abschätzen." Below the text, there are icons for audio, print, and a comment count of 18. At the bottom of the article, there is a large image of a fire.

→ We must control (side-) effects!



Motivation: Security



The screenshot shows a web browser window with the URL <https://www.heise.de/news/Sicherheitslücke-Log4Shell-Internet-in-Flammen>. The article title is "Sicherheitslücke Log4Shell: Internet in Flammen". The text reads: "Die Zero-Day-Sicherheitslücke Log4Shell war zu leicht auszunutzen. Das Ausmaß lässt sich noch immer nicht abschätzen." Below the text, there is a large image of a fire.

- ➔ We must control (side-) effects!
- ➔ disable what is not needed
 - network
 - dynamic loading
 - etc.



Motivation

Experience from using Java in safety-critical systems



Motivation

Experience from using Java in safety-critical systems

→ languages used: **MISRA C**, **Ravenscar Ada**, **Safety-Critical Rust**



Motivation

Experience from using Java in safety-critical systems

- languages used: **MISRA C**, **Ravenscar Ada**, **Safety-Critical Rust**
- need to profit from high-level abstractions



Motivation

Experience from using Java in safety-critical systems

- languages used: **MISRA C**, **Ravenscar Ada**, **Safety-Critical Rust**
- need to profit from high-level abstractions, but
 - Verification & Validation is hard



Motivation

Experience from using Java in safety-critical systems

- languages used: **MISRA C**, **Ravenscar Ada**, **Safety-Critical Rust**
- need to profit from high-level abstractions, but
 - Verification & Validation is hard
 - must not be too dynamic (JIT, dynamic loading, GC, etc.)



Motivation

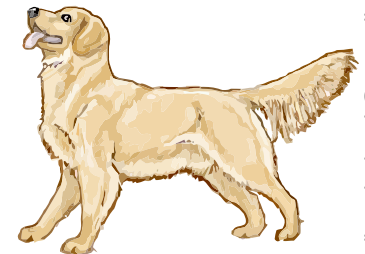
Experience from using Java in safety-critical systems

- languages used: **MISRA C**, **Ravenscar Ada**, **Safety-Critical Rust**
- need to profit from high-level abstractions, but
 - Verification & Validation is hard
 - must not be too dynamic (JIT, dynamic loading, GC, etc.)
- must be open source



Overview

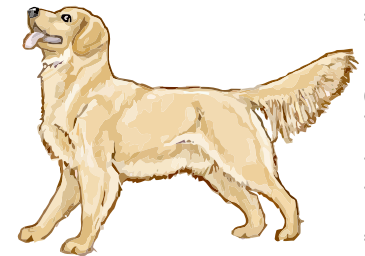
- Motivation
- **Fuzion quick intro**
- Design-by-Contract
- Effects in Fuzion
- Toolchain and Static Analysis
- Safety-Critical Aspects
- Status



clipart by hund @ openciptart.org



Fuzion Intro



Fuzion Intro

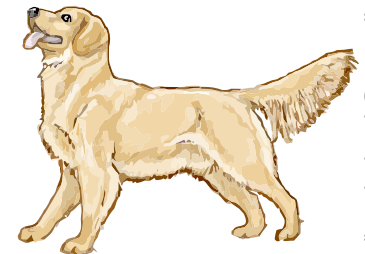
→ One concept: a **feature**

- serves as class / method / function / package / etc.



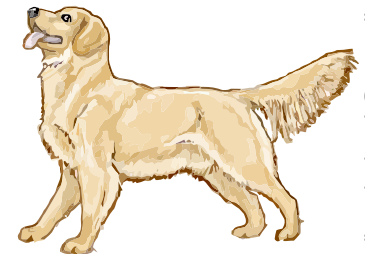
Fuzion Intro

- One concept: a **feature**
 - serves as class / method / function / package / etc.
- Fuzion
 - is statically typed



Fuzion Intro

- One concept: a **feature**
 - serves as class / method / function / package / etc.
- Fuzion
 - is statically typed
 - is polymorphic

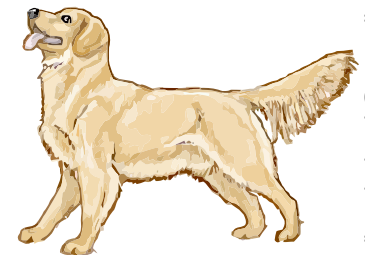


clipart by hund @ openciptart.org



Fuzion Intro

- One concept: a **feature**
 - serves as class / method / function / package / etc.
- Fuzion
 - is statically typed
 - is polymorphic: **union types**

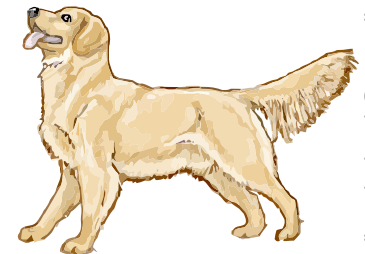


clipart by hund @ opencipart.org



Fuzion Intro

- One concept: a **feature**
 - serves as class / method / function / package / etc.
- Fuzion
 - is statically typed
 - is polymorphic: **union types**, **parametric types**



clipart by hund @ openciptart.org



Fuzion Intro

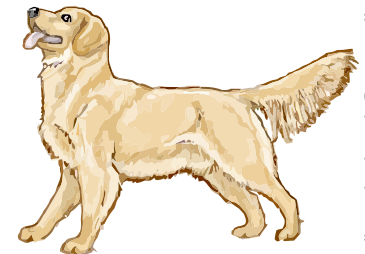
→ One concept: a **feature**

- serves as class / method / function / package / etc.

→ Fuzion

- is statically typed

- is polymorphic: **union types**, **parametric types**, **inheritance**



clipart by hund @ opencipart.org



Fuzion Intro

→ One concept: a **feature**

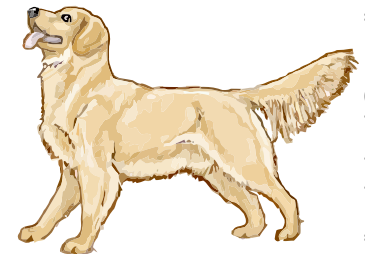
- serves as class / method / function / package / etc.

→ Fuzion

- is statically typed

- is polymorphic: **union types**, **parametric types**, **inheritance**

- uses **design-by-contract**



clipart by hund @ openciptart.org



Fuzion Intro

→ One concept: a **feature**

- serves as class / method / function / package / etc.

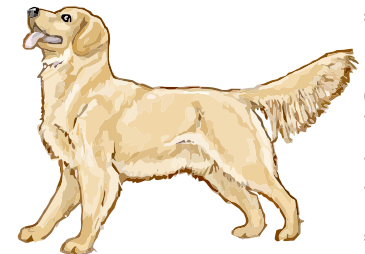
→ Fuzion

- is statically typed

- is polymorphic: **union types**, **parametric types**, **inheritance**

- uses **design-by-contract**

- is pure using **effects**



clipart by hund @ openciptart.org



Fuzion Intro

→ One concept: a **feature**

- serves as class / method / function / package / etc.

→ Fuzion

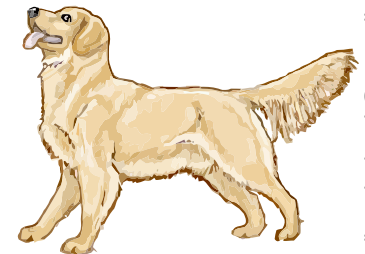
- is statically typed

- is polymorphic: **union types**, **parametric types**, **inheritance**

- uses **design-by-contract**

- is pure using **effects**

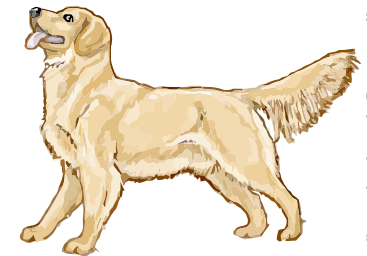
- uses **static analysis**



clipart by hund @ opencipart.org



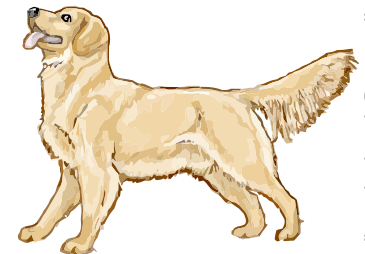
What does Fuzion not have?



What does Fuzion not have?

Capabilities considered harmful:

- Dynamic Loading
- Macros
- Reflection
- Pointer Arithmetic
- (uncontrolled) Mutability
- Exceptions



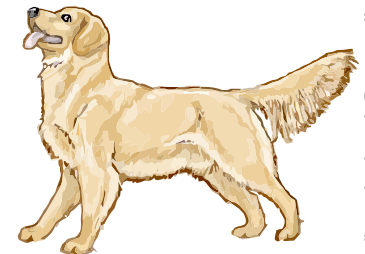
What does Fuzion not have?

Capabilities considered harmful:

- Dynamic Loading
- Macros
- Reflection
- Pointer Arithmetic
- (uncontrolled) Mutability
- Exceptions

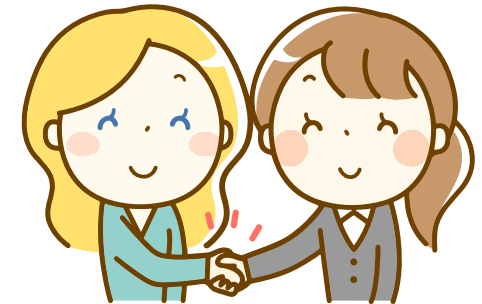
Reasons:

- We must know what code does
- Static Analysis
- Safety
- Performance



Overview

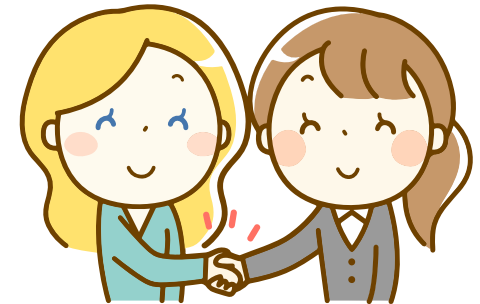
- Motivation
- Fuzion quick intro
- **Design-by-Contract**
- Effects in Fuzion
- Toolchain and Static Analysis
- Safety-Critical Aspects
- Status



clipart by oksmith @ openclipart.org



Design by Contract



clipart by oksmith @ openclipart.org



Design by Contract

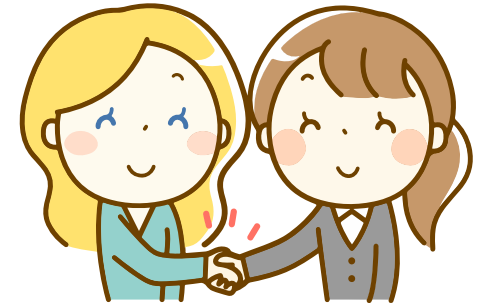
Nothing new

→ concept presented by Bertrand Meyer back in 1986

Features define their behavior

→ pre-condition: what has to hold before a call?

→ post-condition: what guarantee is given after the call?

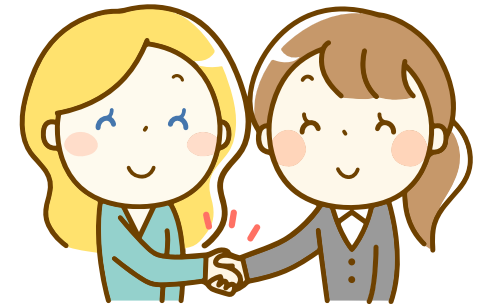


clipart by oksmith @ openclipart.org



Design by Contract: Example

```
sqrt(a I : integer) I
  pre
    a >= 0
  post
    result * result <= a < (result+1) * (result+1)
=>
...
```



clipart by oksmith @ openclipart.org



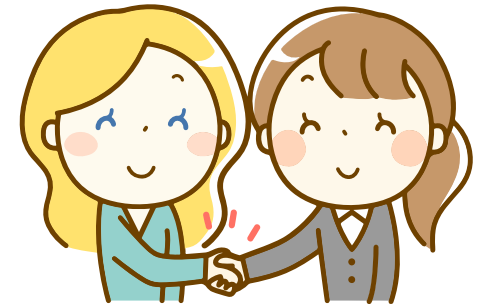
Controlling Contract Checks

Checking contracts dynamically

- will introduce run-time overhead
- may be prohibitively expensive
- may be required for safety

Solution

- qualified contracts

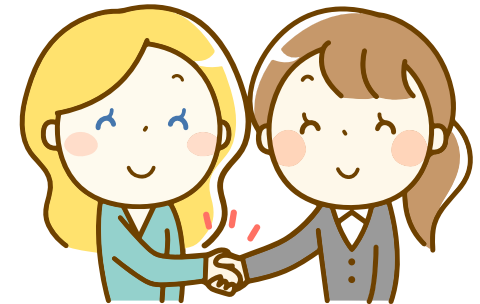


clipart by oksmith @ openclipart.org



Qualified Contracts

```
sqrt(a N : numeric) N
  pre
    debug: a >= 0
  post
    debug 5: result * result <= a < (result+1) * (result+1)
=>
...
```



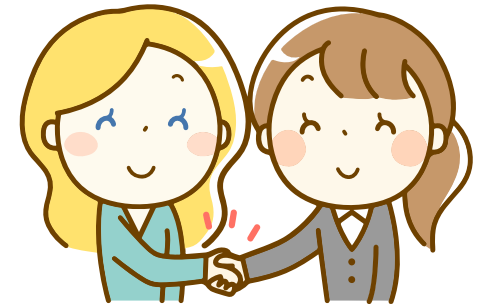
clipart by oksmith @ openclipart.org



Contract Qualifiers

Fuzion contract qualifiers

- **safety**
- **debug**
- **debug n**
- **pedantic**
- **analysis**



clipart by oksmith @ openclipart.org

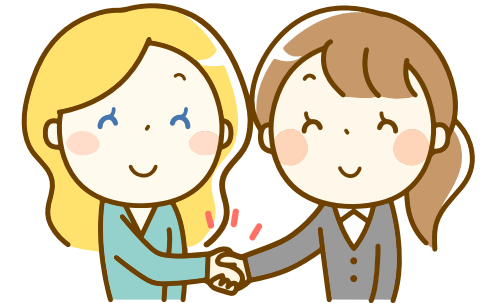


Contracts for Static Analysis

```

max(s Sequence (T : property.orderable)) T
  pre
    debug: !s.isEmpty
  post
    debug: s ∀ (<= result)
    debug: s ∃ (= result)
    analysis: for_all T (x -> s.contains x: x <= result)
=>
...

```



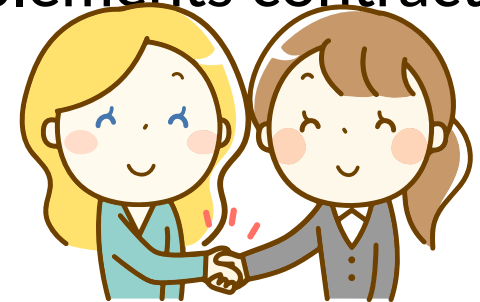
clipart by oksmith @ openclipart.org



Design-by-Contract & Certification

Contracts provide

- direct way to add formal requirements to code
- means to verify these requirements at runtime
- means to define (or generate) tests
- formal analysis tools may verify code implements contract



clipart by oksmith @ openclipart.org



Overview

- Motivation
- Fuzion quick intro
- Design-by-Contract
- **Effects in Fuzion**
- Toolchain and Static Analysis
- Safety-Critical Aspects
- Status



clipart by magicbraingames @ openclipart.org



Effects

Encapsulate non-functional aspects

- input / output
- resources: time, memory
- non-local control flow (exceptions, coroutines, aborting, etc.)



clipart by magicbraingames @ openclipart.org



Effects

Definition

- an effect is a set of operations
 - **read**, **get_time**, **panic**, **log**, ...
 - operations often model a non-functional aspect
- operations may **resume** or **abort**
- an effect is implemented by a **handler**
- to use effect operations, a handler must be instated



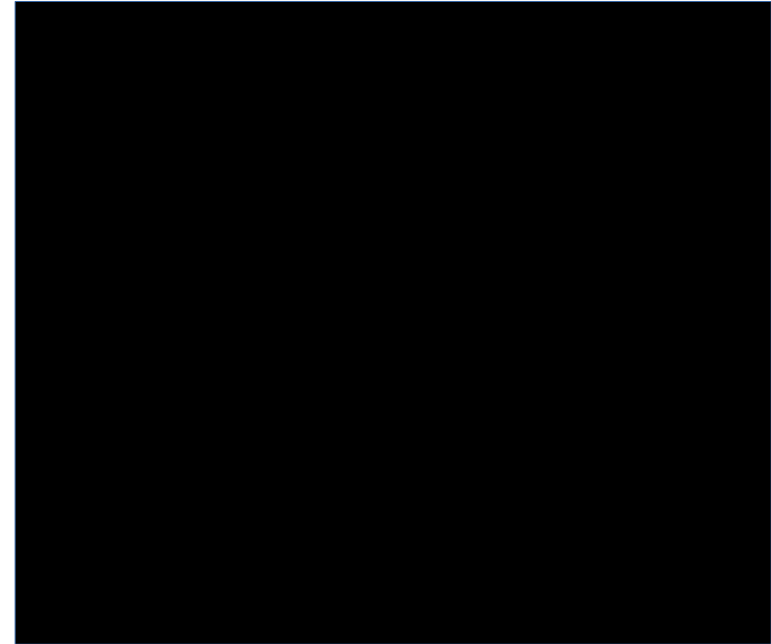
clipart by magicbraingames @ opencart.org



Fuzion Effects Example

```
hello_world ! io.Out =>  
  io.Out.env.println "hello world!"
```

```
hello_world
```



Fuzion Effects Example

```
hello_world ! io.Out =>  
  io.Out.env.println "hello world!"
```

```
hello_world
```

```
> fz hw.fz
```



Fuzion Effects Example

```
hello_world ! io.Out =>  
  io.Out.env.println "hello world!"
```

```
hello_world
```

```
> fz hw.fz  
hello world!  
>
```



Fuzion Effects Example

```
hello_world ! io.Out =>  
  io.Out.env.println "hello world!"
```

```
hello_world
```

```
> fz hw.fz  
hello world!  
> fz -effects hw.fz
```



Fuzion Effects Example

```
hello_world ! io.Out =>  
  io.Out.env.println "hello world!"
```

```
hello_world
```

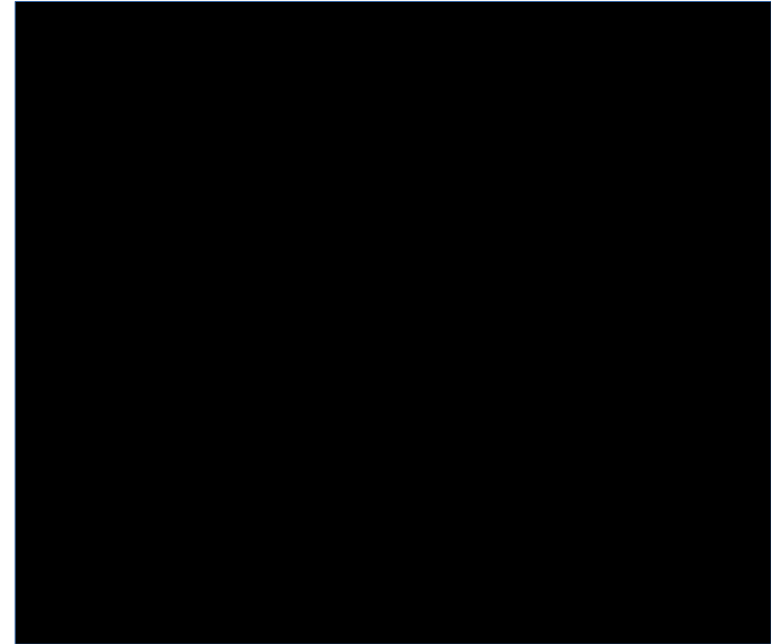
```
> fz hw.fz  
hello world!  
> fz -effects hw.fz  
io.Out  
[ ... ]  
>
```



Fuzion Effects Example

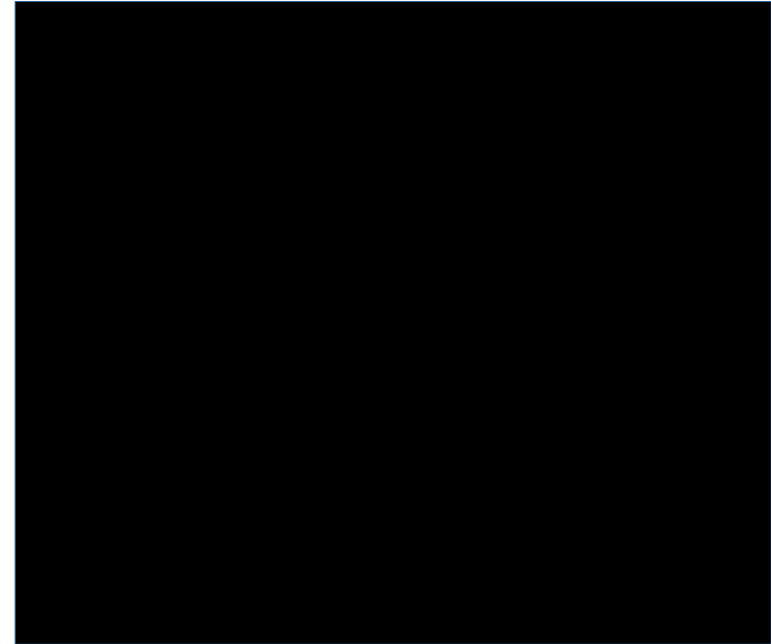
```
hello_world ! io.Out =>  
  io.Out.env.println "hello world!"
```

```
my_handler : io.Out is
```



Fuzion Effects Example

```
hello_world ! io.Out =>  
  io.Out.env.println "hello world!"  
  
my_handler : io.Out is  
  public redef print(s Any) unit =>
```



Fuzion Effects Example

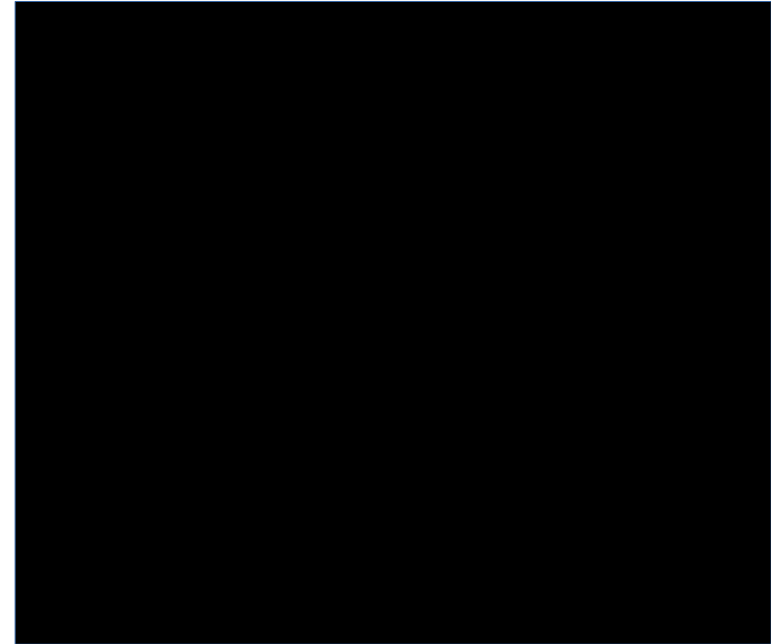
```
hello_world ! io.Out =>
```

```
  io.Out.env.println "hello world!"
```

```
my_handler : io.Out is
```

```
  public redef print(s Any) unit =>
```

```
    io.err.print (($s).replace "!" "!!!11!")
```



Fuzion Effects Example

```
hello_world ! io.Out =>
```

```
  io.Out.env.println "hello world!"
```

```
my_handler : io.Out is
```

```
  public redef print(s Any) unit =>
```

```
    io.err.print (($s).replace "!" "!!!!11!")
```

```
io.Out.instate my_handler ()→
```

```
  hello_world
```



Fuzion Effects Example

```
hello_world ! io.Out =>
```

```
  io.Out.env.println "hello world!"
```

```
my_handler : io.Out is
```

```
  public redef print(s Any) unit =>
```

```
    io.err.print (($s).replace "!" "!!!11!")
```

```
io.Out.instate my_handler ()→
```

```
  hello_world
```

```
> fz hw.fz
hello world!!!11!
>
```



Fuzion Effects Example

```
hello_world ! io.Out =>
```

```
  io.Out.env.println "hello world!"
```

```
my_handler : io.Out is
```

```
  public redef print(s Any) unit =>
```

```
    io.err.print (($s).replace "!" "!!!11!")
```

```
io.Out.instate my_handler ()→
```

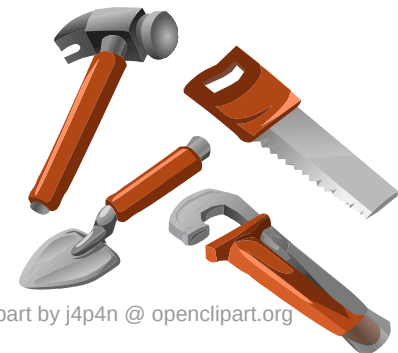
```
  hello_world
```

```
> fz hw.fz
hello world!!!11!
> fz -effects hw.fz
io.Err
[ ... ]
>
```



Overview

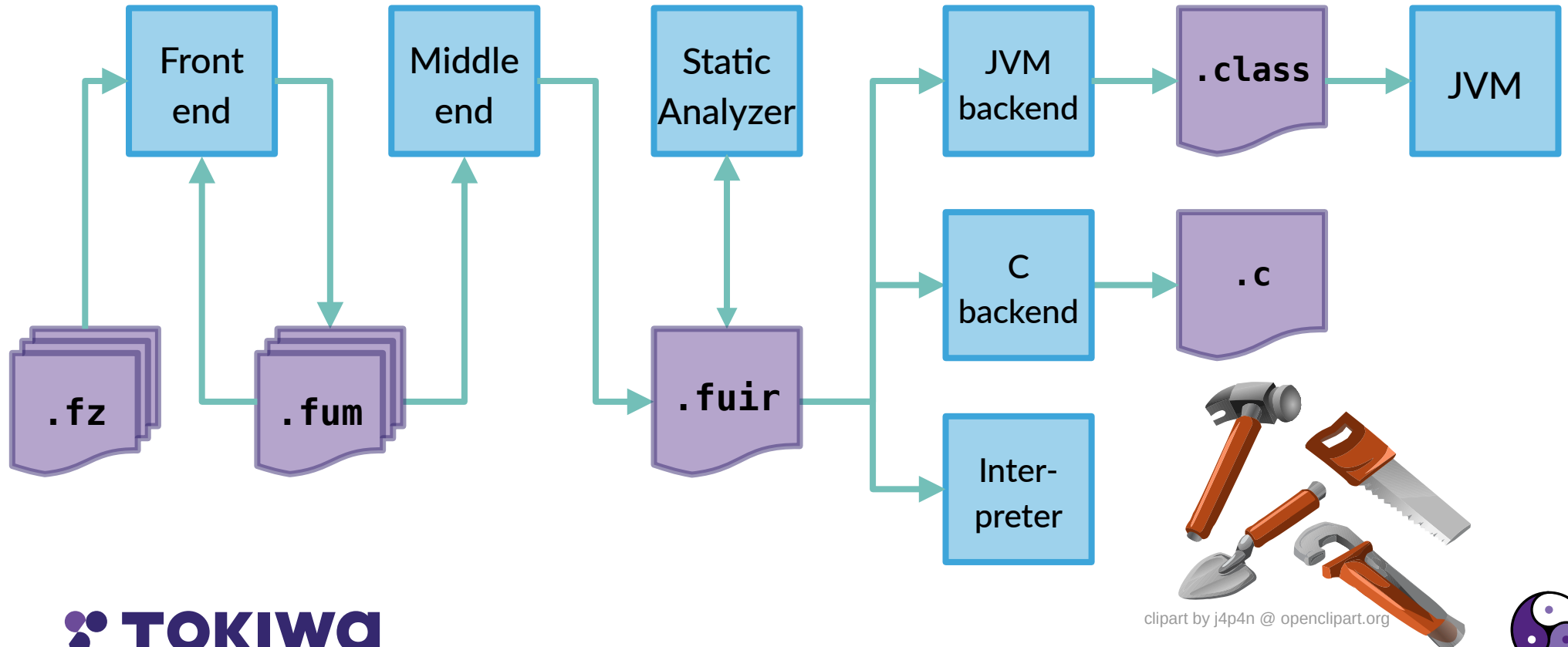
- Motivation
- Fuzion quick intro
- Design-by-Contract
- Effects in Fuzion
- **Toolchain and Static Analysis**
- Safety-Critical Aspects
- Status



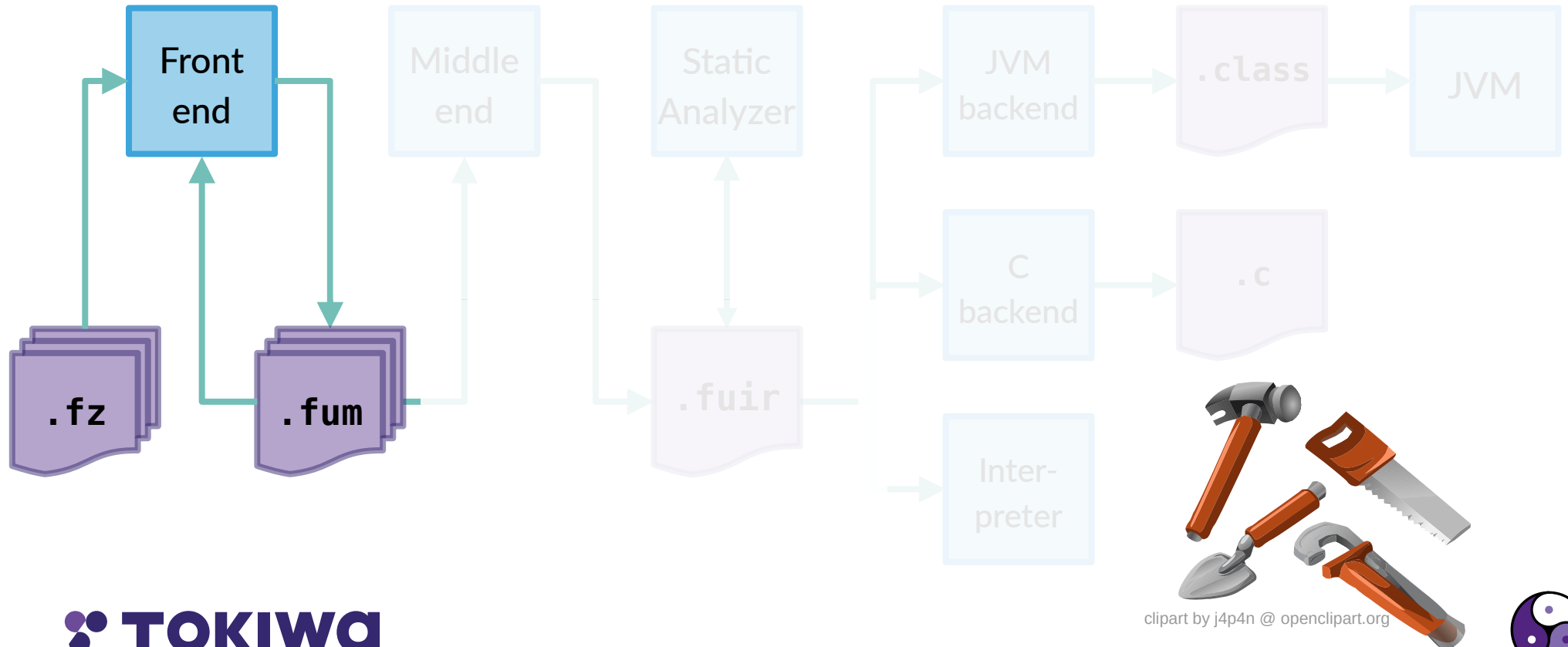
clipart by j4p4n @ openclipart.org



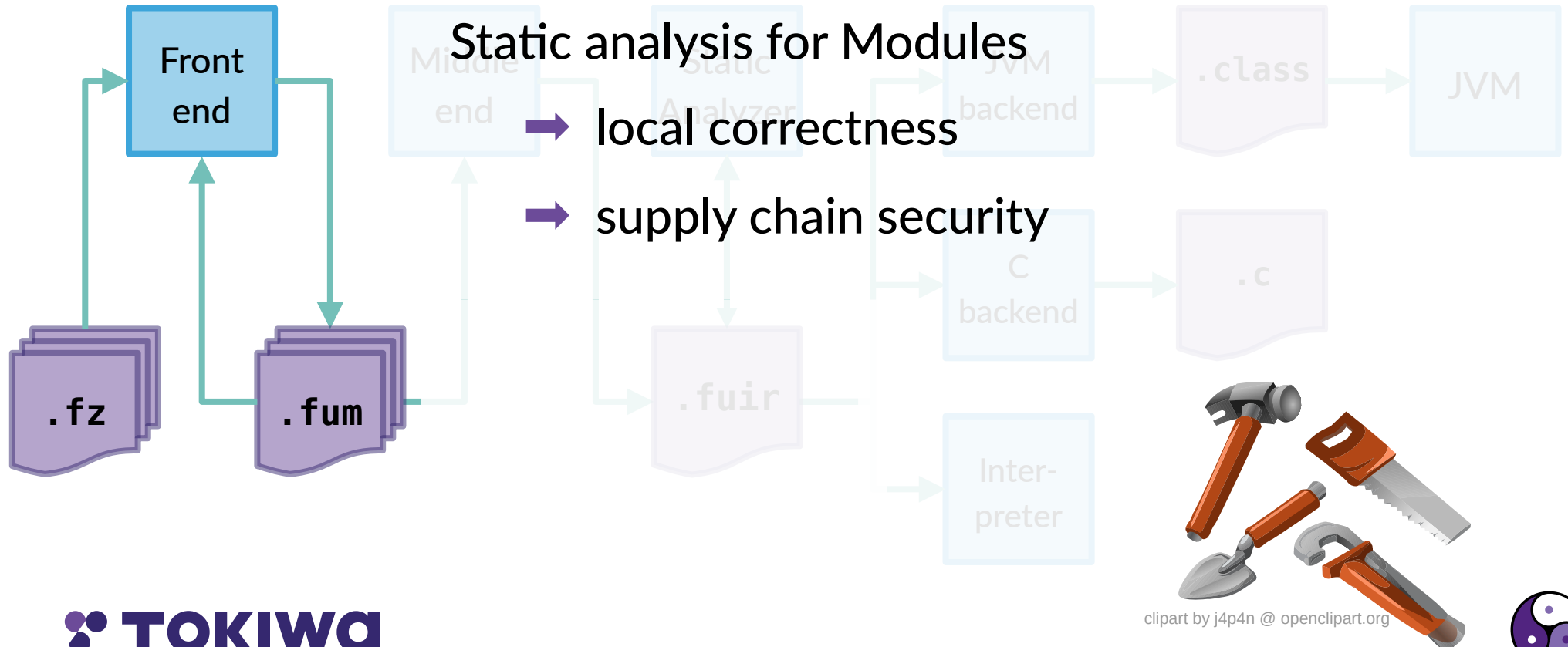
Fuzion Toolchain and Static Analysis



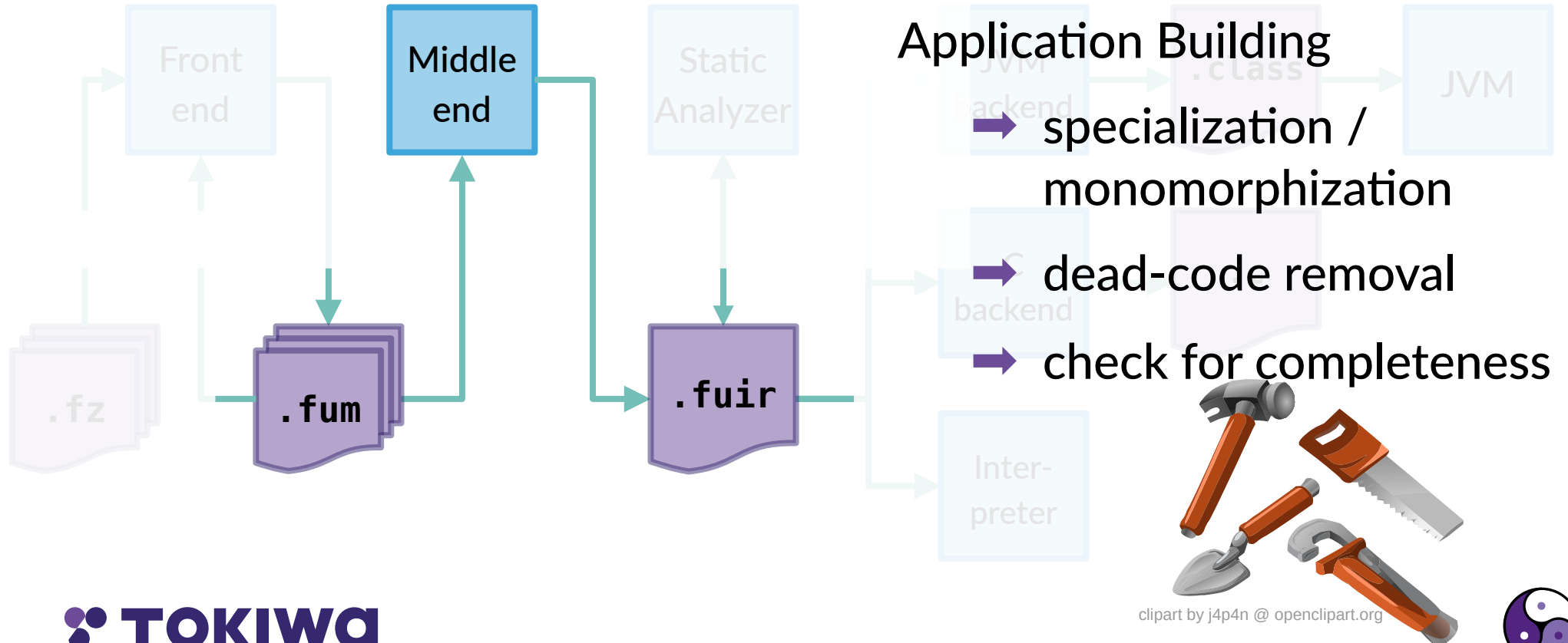
Fuzion Toolchain and Static Analysis



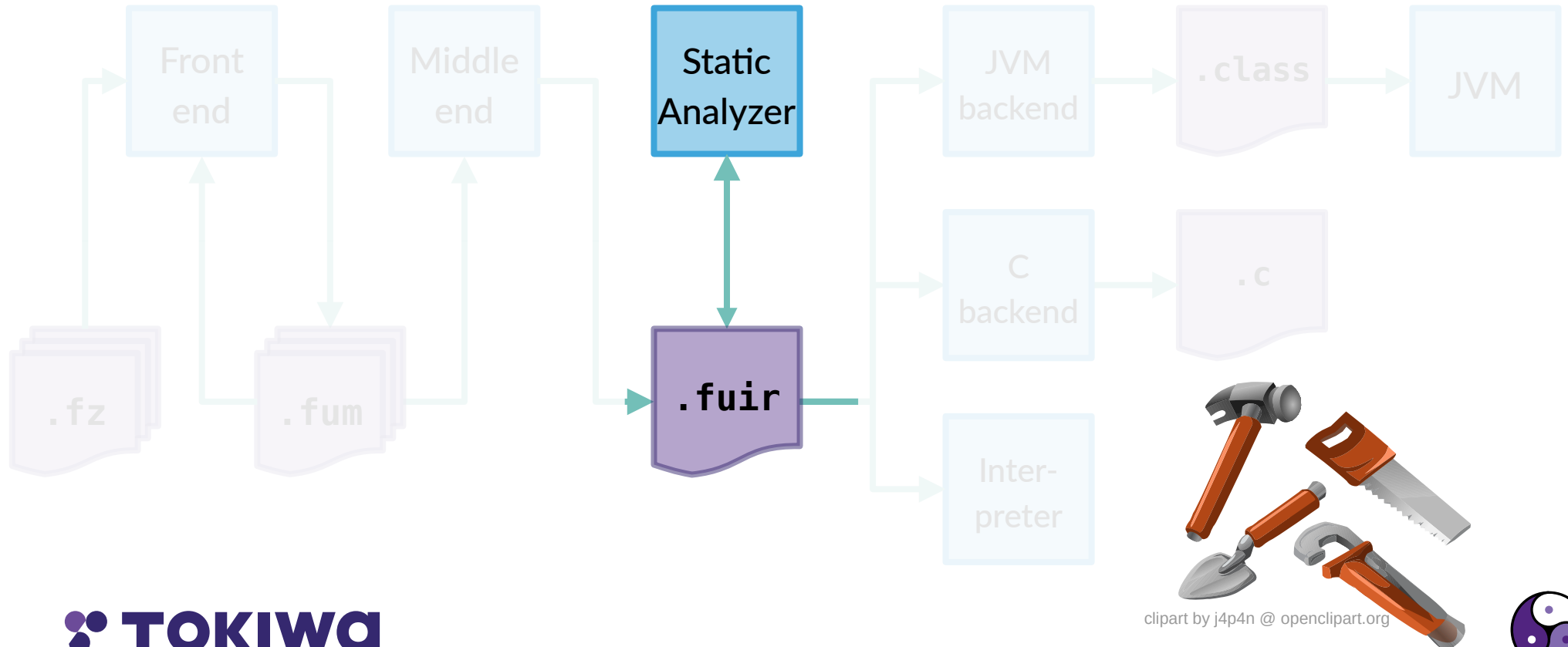
Fuzion Toolchain and Static Analysis



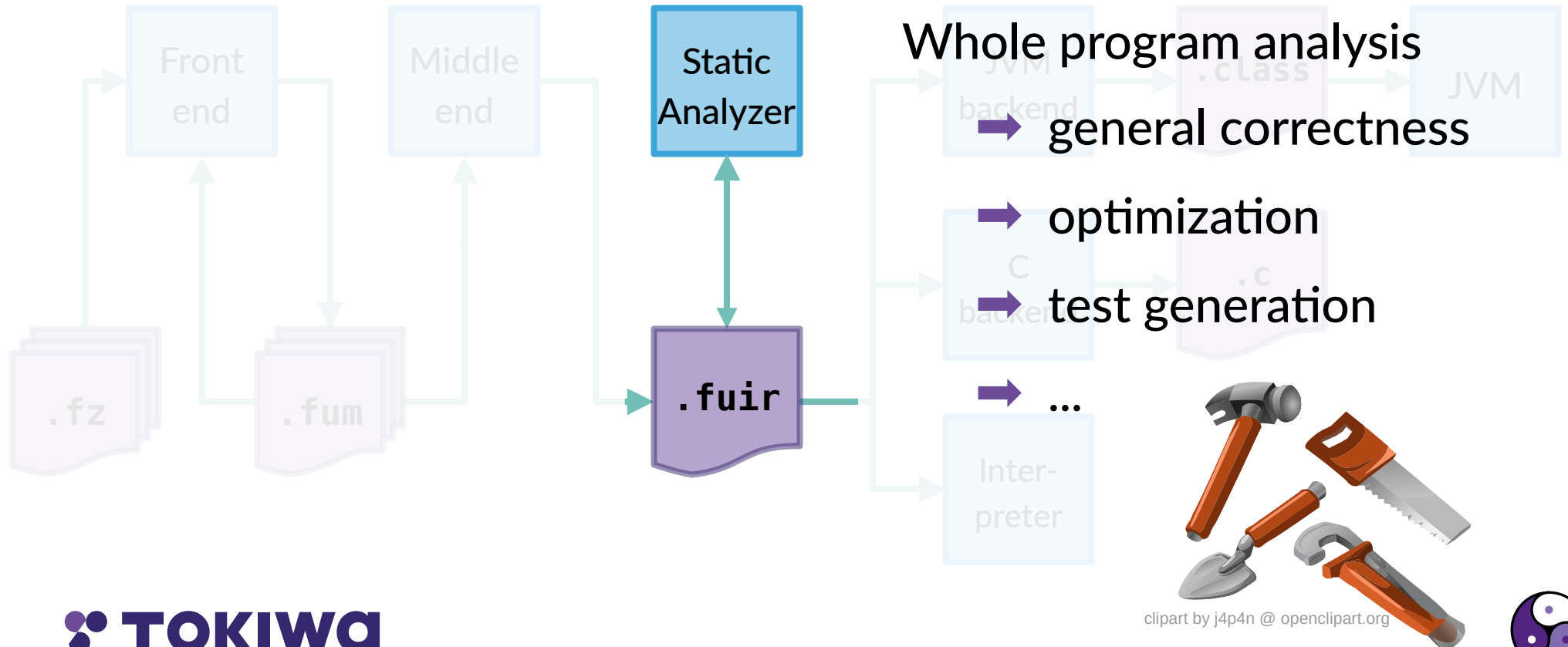
Fuzion Toolchain and Static Analysis



Fuzion Toolchain and Static Analysis

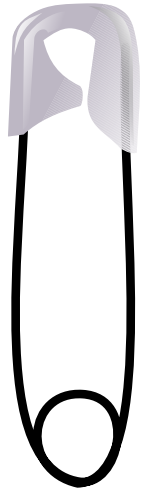


Fuzion Toolchain and Static Analysis



Overview

- Motivation
- Fuzion quick intro
- Design-by-Contract
- Effects in Fuzion
- Toolchain and Static Analysis
- **Safety-Critical Aspects**
- Status



clipart by Anonmous @ openclipart.org



Safety-Critical Aspects

Design-by-contract for

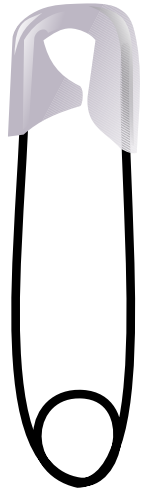
- verification & validation, e.g. test generation

Effects for

- supply chain security: limit functionality in library code
- resource limitations / real-time tasks

Static Analysis

- correctness proofs as certification artifacts



clipart by Anonmous @ openclipart.org



Overview

- Motivation
- Fuzion quick intro
- Design-by-Contract
- Effects in Fuzion
- Toolchain and Static Analysis
- Safety-Critical Aspects
- **Status**



Fuzion: Status

- First real-world application: fzweb, fuzion tools, etc.
- ITEA4 project greencode.ai:
 - real-time APIs
 - static analysis
 - runtime tools
- JVM and C backends
- Basic static analysis available



Thank you. Any questions?

Please follow and stay informed

- github.com/tokiwa-software/fuzion
- fuzion-lang.dev
- [@fuzionlang.bsky.social](https://bsky.app/profile/fuzionlang.bsky.social)
- [@fuzion@types.pl](mailto:fuzion@types.pl)

Contact us!

