SAMSUNG
Open Source Group

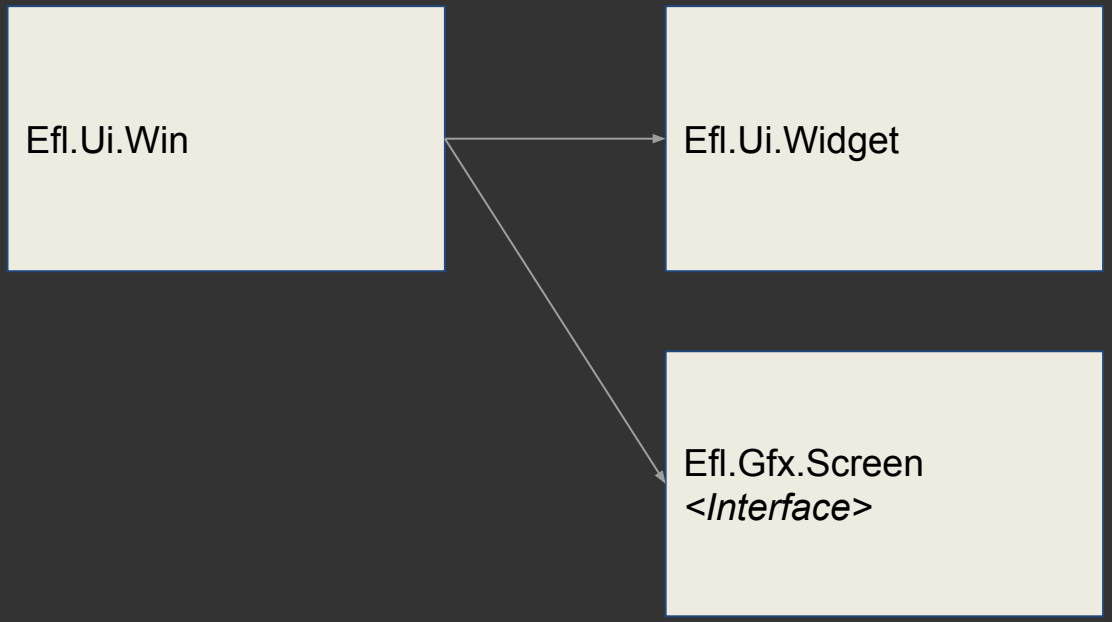
# Eo Performance optimization

# Basic assertions

- Class inheritance is **stable** (does not change in runtime)
  - The **parent** class is always known
  - The **extension** classes are always known
  - Privat data structs are in linear order

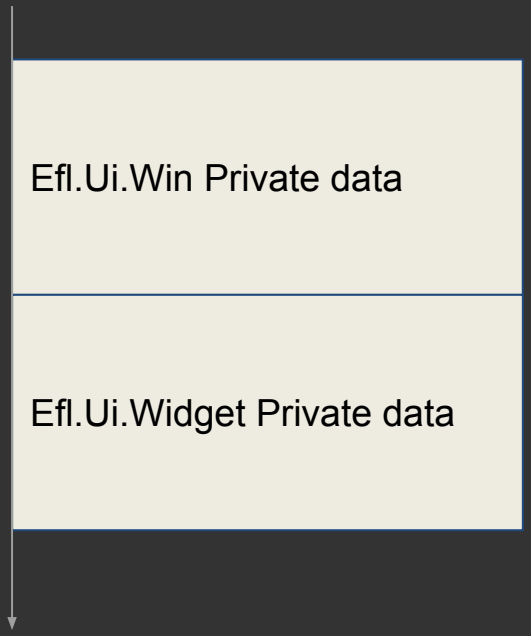
# Basic assertions

- Class inheritance is **stable** (does not change in runtime)



# Basic assertions

- Privat data structs are in linear order



# Lemmas

- The conceptional result of efl_super(obj, MY_CLASS) is **const** and calculable in **compiletime** (For regular and abstract implementations)

# Example

- Window object
- Setting a **position** (Efl.Gfx.Entity.position)

- Implementing entities:
  - Efl.Ui.Win
  - Efl.Ui.Widget
  - Efl.Canvas.Group
  - Efl.Canvas.Object

# Example (2)

- That means:
- 4 Times **decoding** the same eo_id
- 4 Times **calculating** the private data ptr
- 4 Times **resolving** the calling function

# Example (3)

- Decomposing eo_id is only needed to access **vtable**
- Knowning the initial **private data** pointer makes every **private data** pointer easy calculatable
- The result of the **resolving** is known for regular / abstracts

# Example (3)

## Normal code in Efl.Ui.Widget:

```
EOLIAN static void
_efl_ui_widget_efl_gfx_entity_position_set(Eo *obj, Elm_Widget_Smart_Data *sd, Eina_Position2D pos){
   [...]
   efl_gfx_entity_position_set(efl_super(obj, MY_CLASS), pos);
}
```

# What we can do about it

- In a structure like :

```
Y_X(obj, ...) {
   [...]

X(efl_super(obj,C),...)
   [...]
}
```

- We can replace:

**X(efl_super(obj, C), ...)**

with

X'(obj, pd + diff, ...)

# What we can do about it

- **X'** is the function that is implemented for X in the next lower level to Y.
- **Diff** is the ptr difference for making the current private data, the private data required for calling X'.

# How can we do that ?

- Write a **gcc/clang** plugin
- Make eolian more expressive (We need the implemented methods of classes, across the tree)
- Replace **X(efl_super(obj, MY_CLASS), ...)** calls

# How much will it improve?

| App | call_resolve | efl_super | % |
|---|---|---|---|
| enlightenment | 1430520 | 164089 | 11,47% |
| terminology | 732699 | 81363 | 11,10% |
| evisum | 3344546 | 303432 | 9,07% |
| efl-netctl | 1047363 | 114949 | 10,98% |
| radio example | 83567 | 10799 | 12,92% |

- Those are theoretical numbers (including mixins), the real safe up will be **lower.**
- This also has a impact to CPU utilization etc. (which is hard to predict)

# How do we run it

- **Build** the plugin **intree**

- Add flags to internal CFLAGS

- **No** external dependency
- **No** additional installing

SAMSUNG
Open Source Group

# Questions ?