Compile-time analysis:

```
//type safe IPC compile-time analysis
//implemented as a compiler pass at the MIR stage
basic_blocks = MIR.get_basic_blocks() //built-in compiler fxn
for block in basic_blocks:
    //send analysis
    if block is a function call with the name SEND_FXN:
        in = input variable to the fxn call
        original_def = original definition of in
        if original_def was a transmute:
            in = input variable to the transmute
            original_def = original definition of in
        type_info = get_type(original_def)
        insert_ipc_send(block, hash(type_info))
    //receive analysis
    if block is a function call with the name RECEIVE_FXN:
        out = variable that stores the received data
        first_use = first use of variable out
        if first_use is a transmute:
            out = variable that stores the output of the transmute
            first_use = first use of variable out
        type_info = get_type(first_use)
        insert_ipc_receive(block, hash(type_info))
```

Runtime library:

```
//type safe IPC runtime library

fn send_type(actual_type, CHANNEL_ID):
    type_channel = get_channel(CHANNEL_ID)
    type_channel.send(actual_type)

fn receive_type(expected_type, CHANNEL_ID):
    type_channel = get_channel(CHANNEL_ID)
    actual_type = type_channel.receive()
    assert_eq!(actual_type, expected_type)
```