



PUSHING THE LIMITS OF LABVIEW

# **Beyond State Machines:**

## **Building Modular Applications in LabVIEW Using Public & Private Events**

Justin Goeres

Senior Engineer & Product Marketing Manager

JKI

# Rules

1. Vote with your feet.

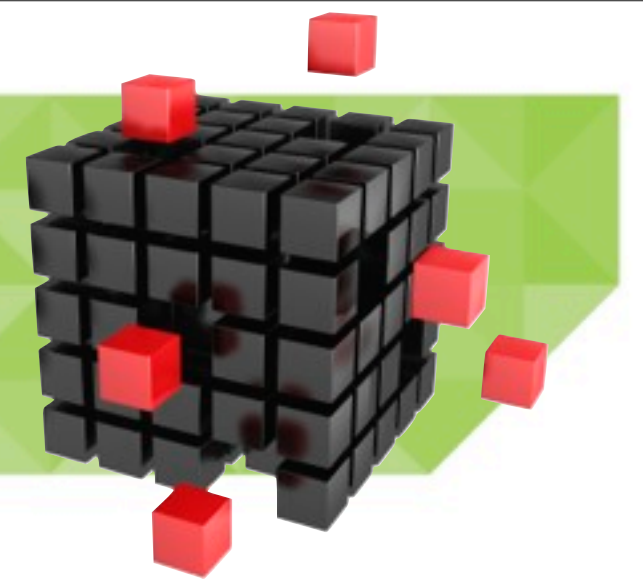


2. Please **turn on** your smartphones.



[@JustinGoeres](#) is going Beyond State Machines at [#niweek](#)!

# Let's Get This Straight



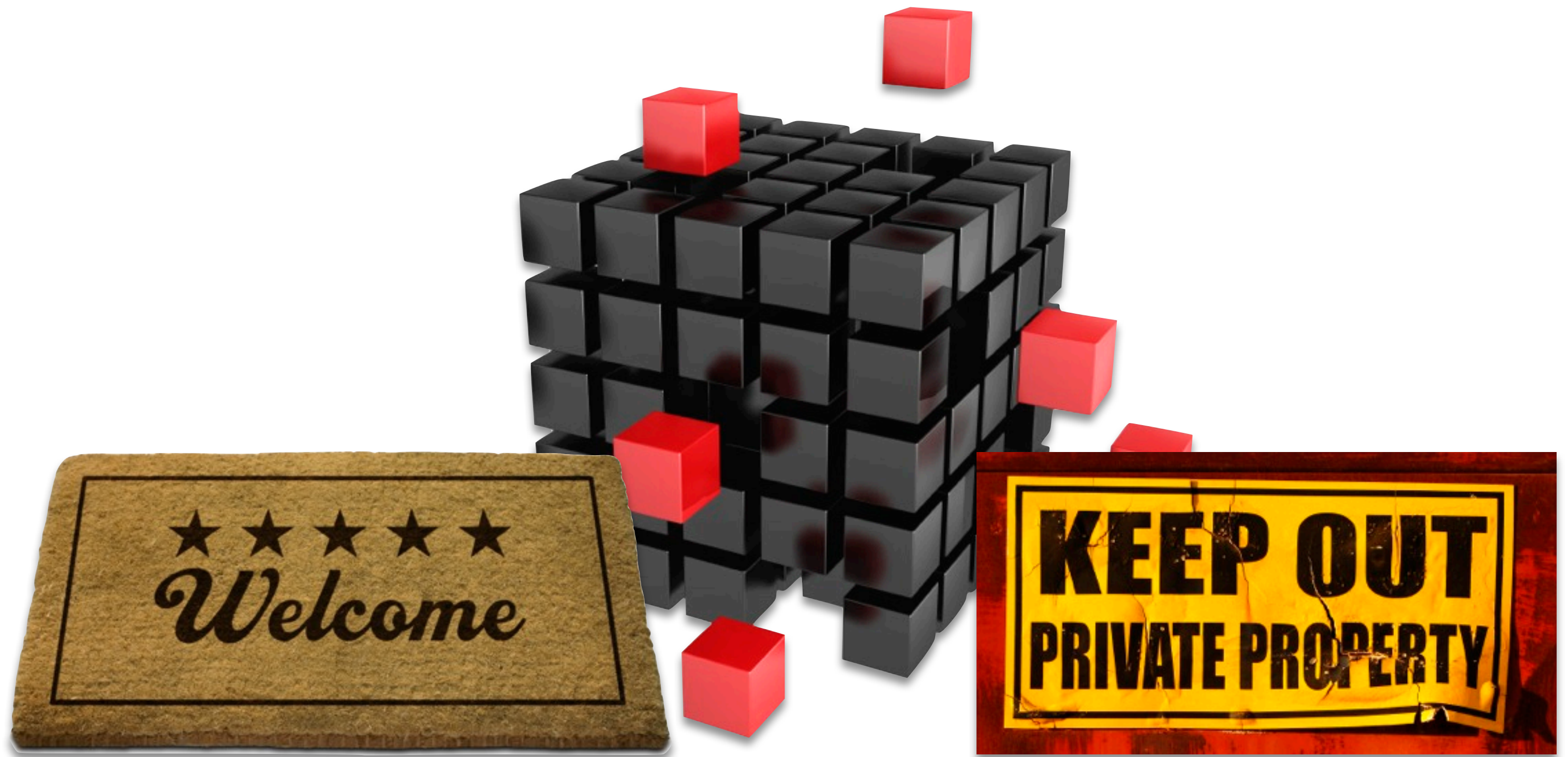
User Events are a very easy-to-use feature with a lot of cool functionality.

They form the basis of JKI's primary application frameworks & templates.

If we could get a couple things fixed/added to LabVIEW, we could do even better.

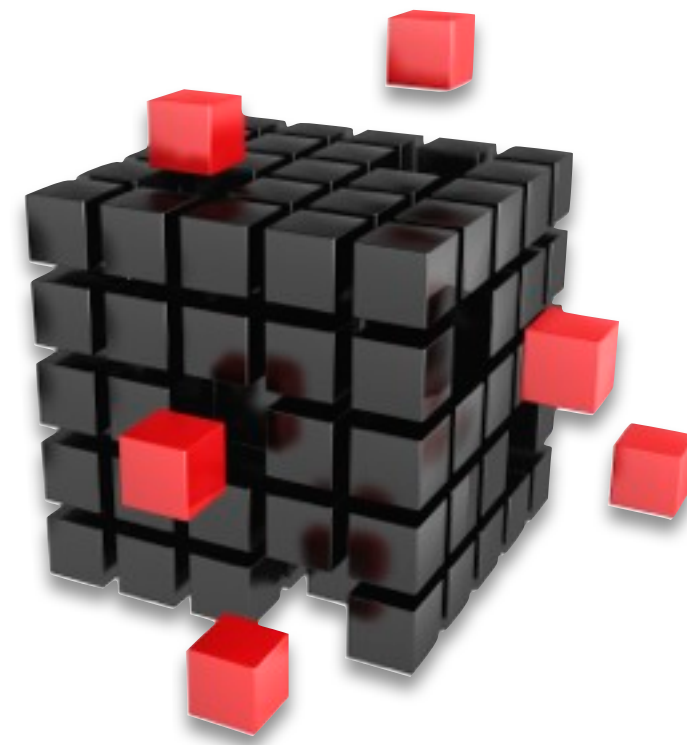


# A Public & Private Events Framework



# Agenda

## What is it?



# Agenda

What is it?

How does it work?



# Agenda

What is it?

How does it work?

What's it good for?





# Agenda

What is it?

How does it work?

What's it good for?

What's it bad at?





# Agenda

What is it?

How does it work?

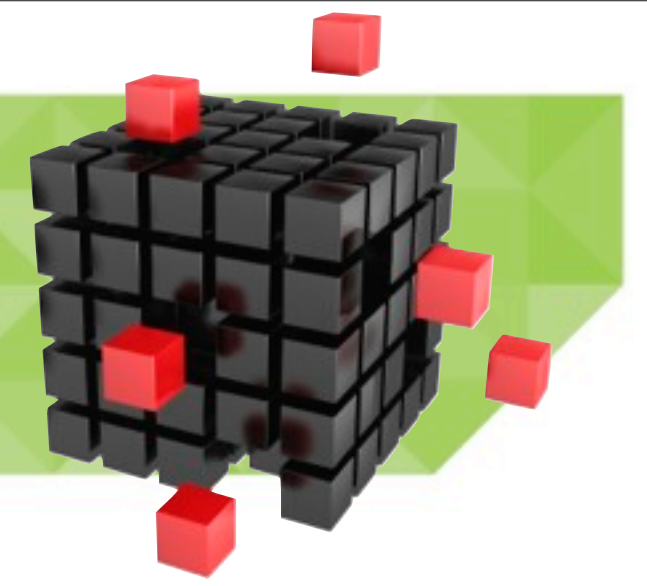
What's it good for?

What's it bad at?

How can you use it in your projects?



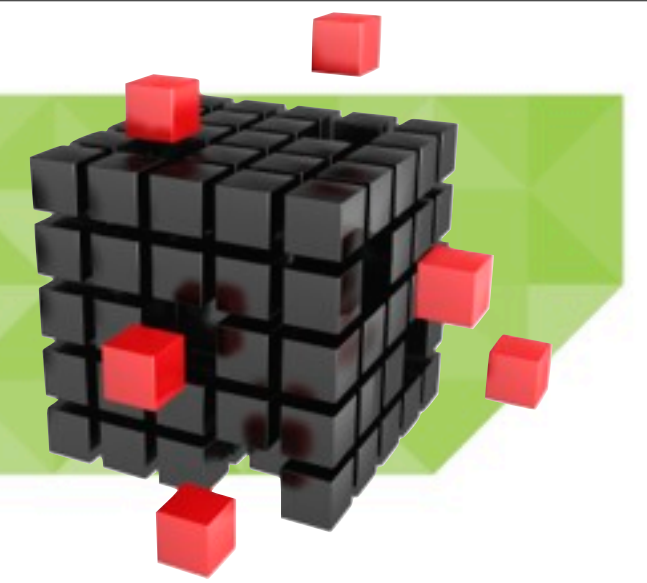
# What is it?



A “module” has:

- Statefulness / private data
- Asynchronous process(es)
- Public API
- Public Events

# Key Benefits

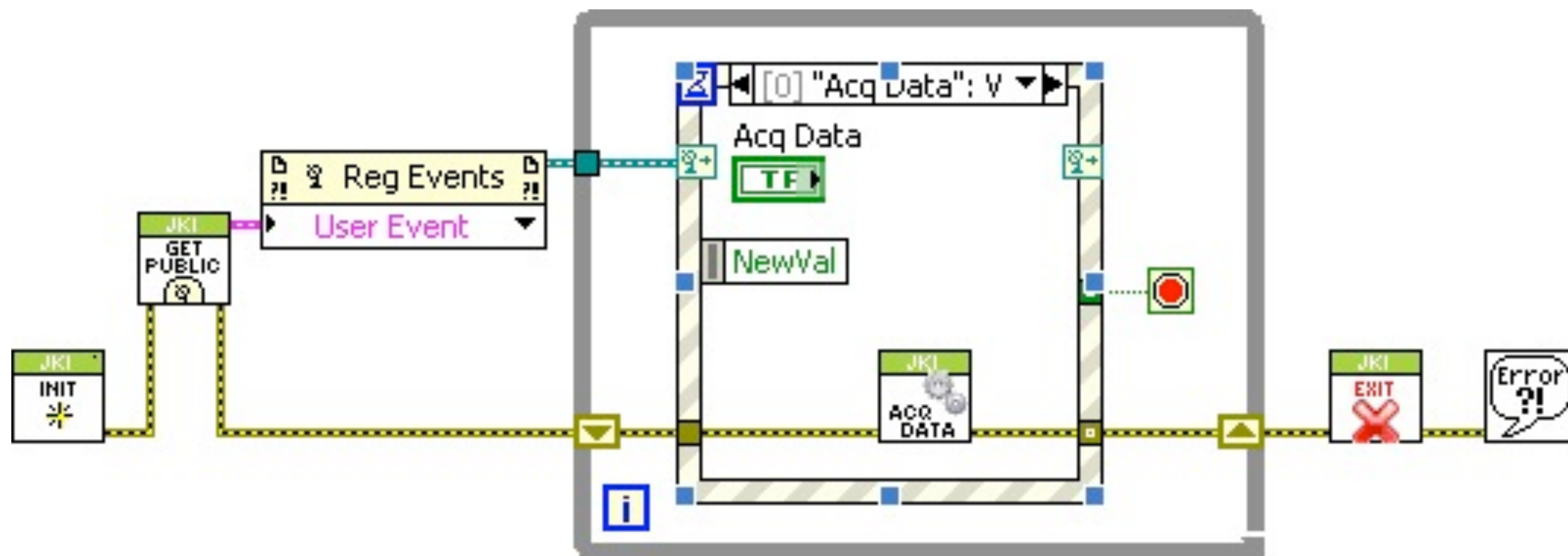


Module knows nothing about who's consuming the data it produces

Lightweight & easy to use

Compatible with by-ref and by-val architectures

# Client Code is Simple



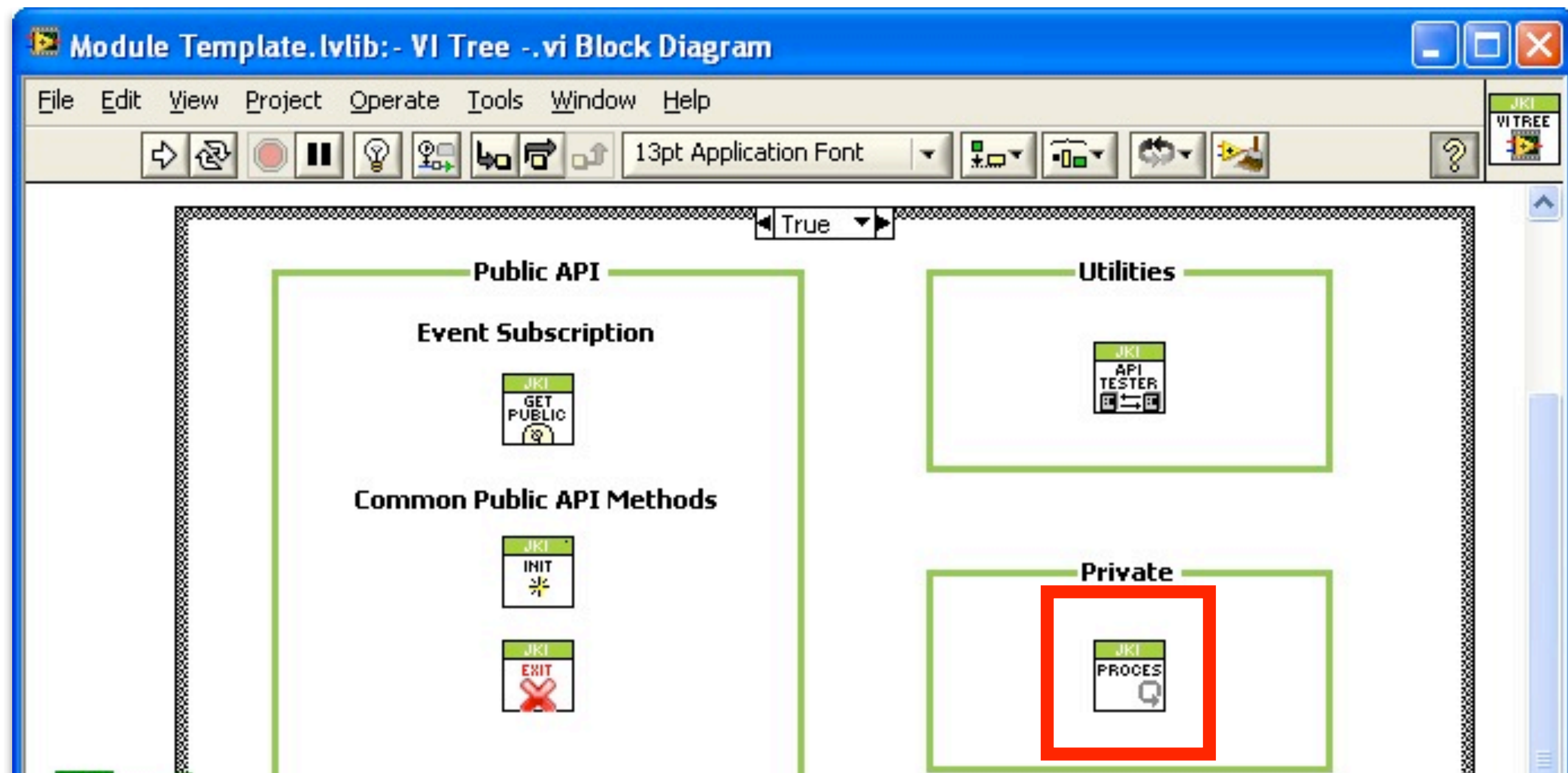
Start Process &  
register for Public  
Events

Invoke Public API  
as needed

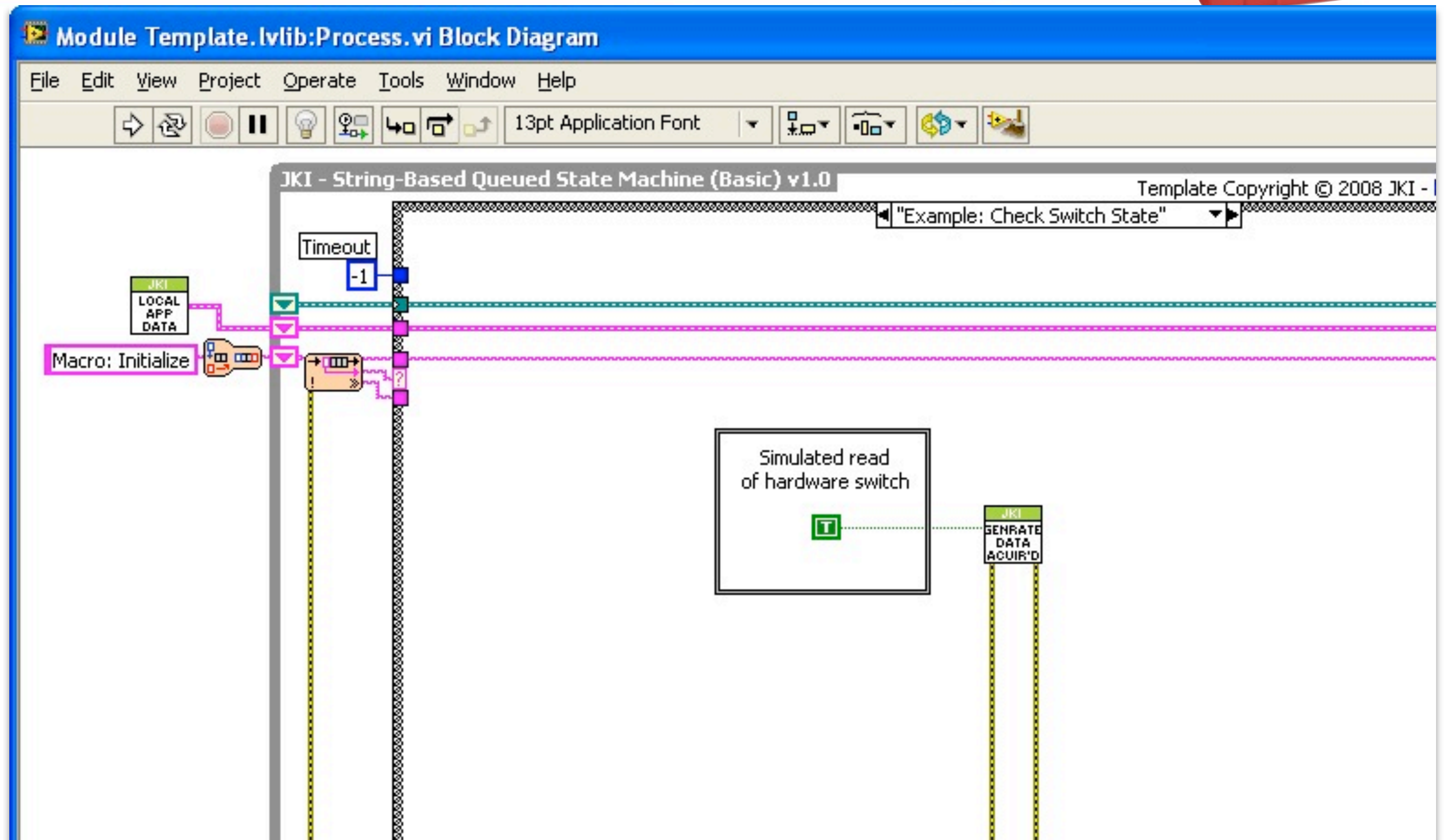
Exit Module  
when done



# How Does it Work?

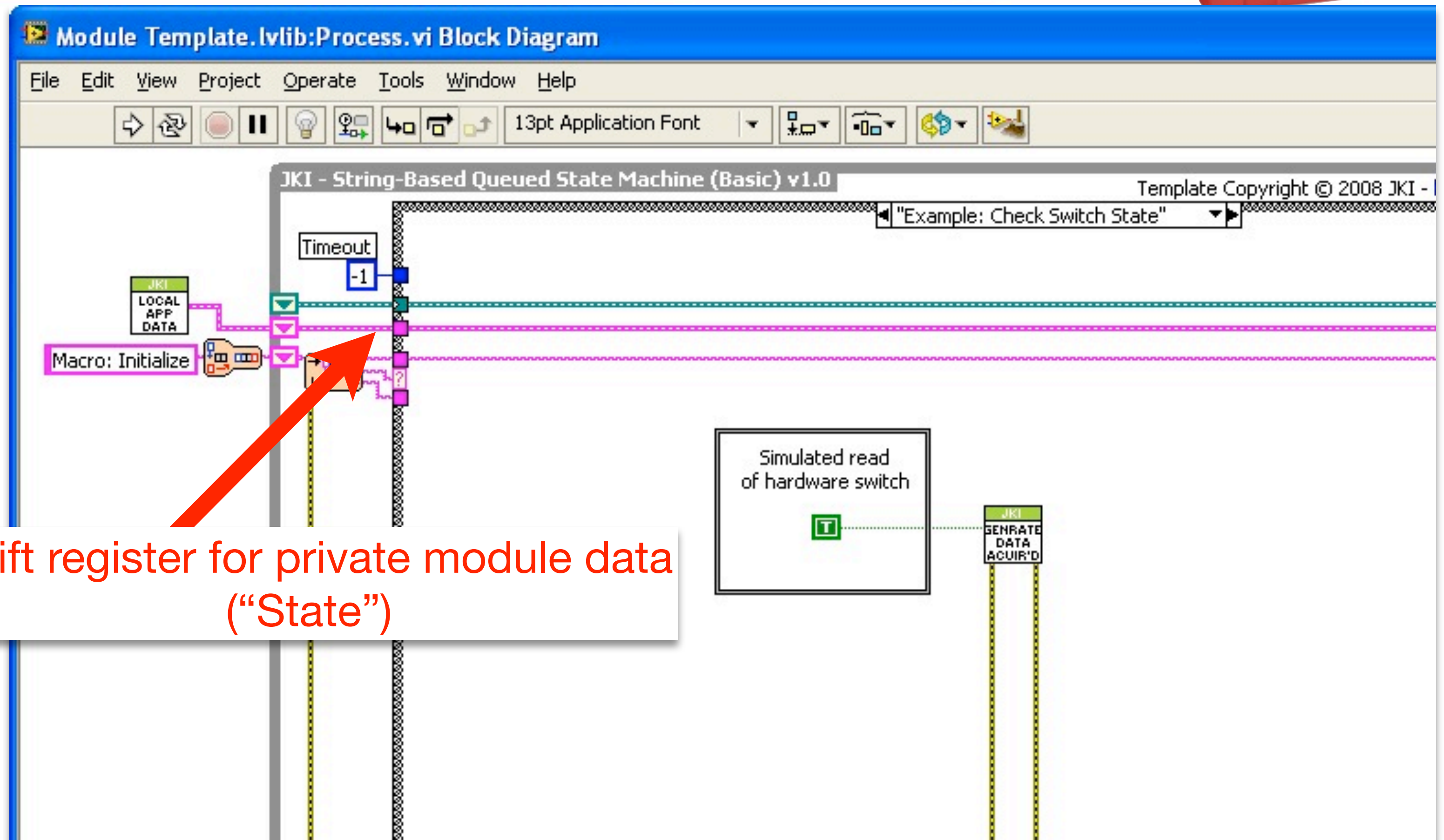


# Process VI & Public Events



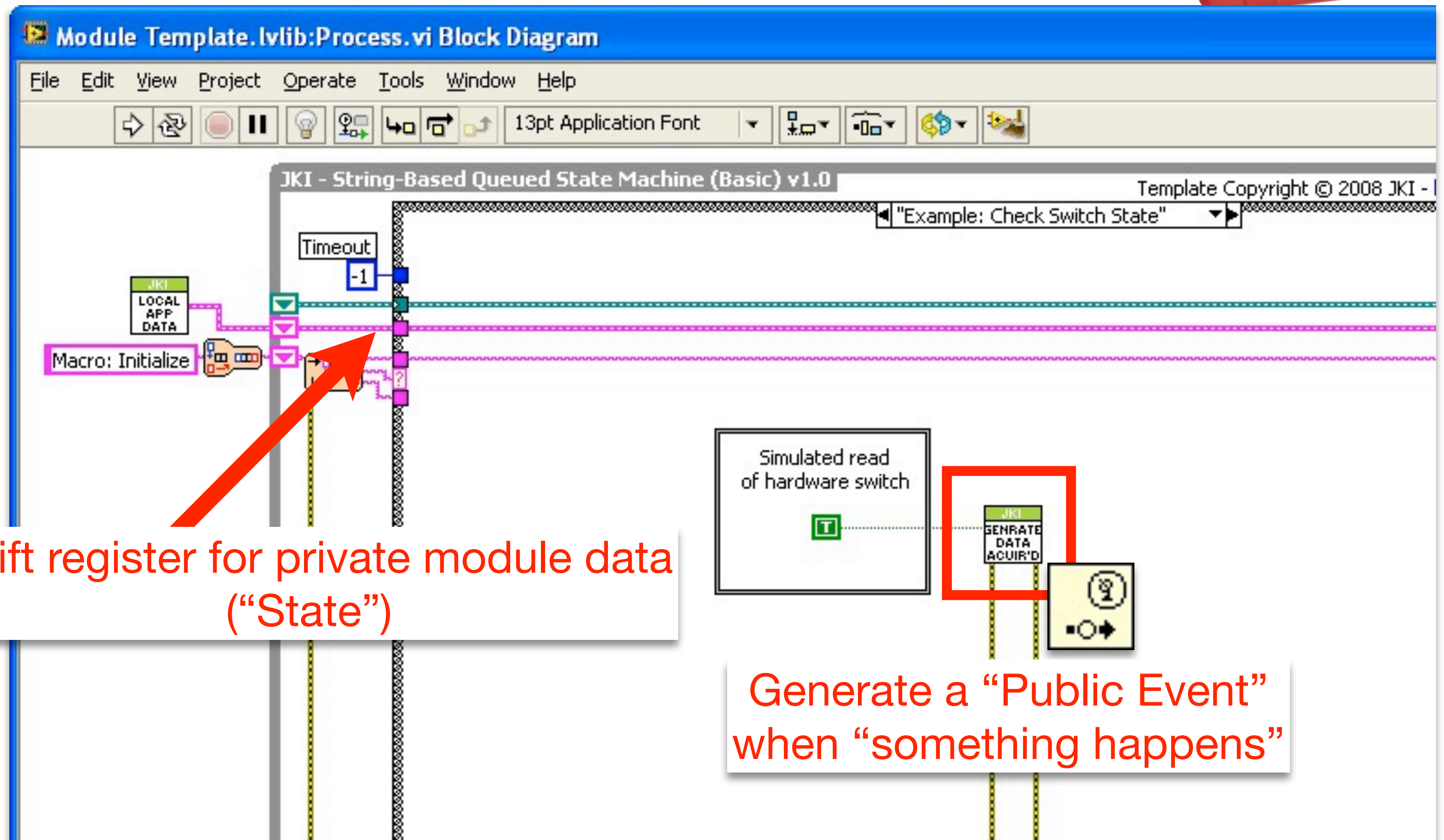


# Process VI & Public Events



Shift register for private module data  
("State")

# Process VI & Public Events



Shift register for private module data  
("State")

Generate a "Public Event"  
when "something happens"



# Public Events: Sending Information OUT



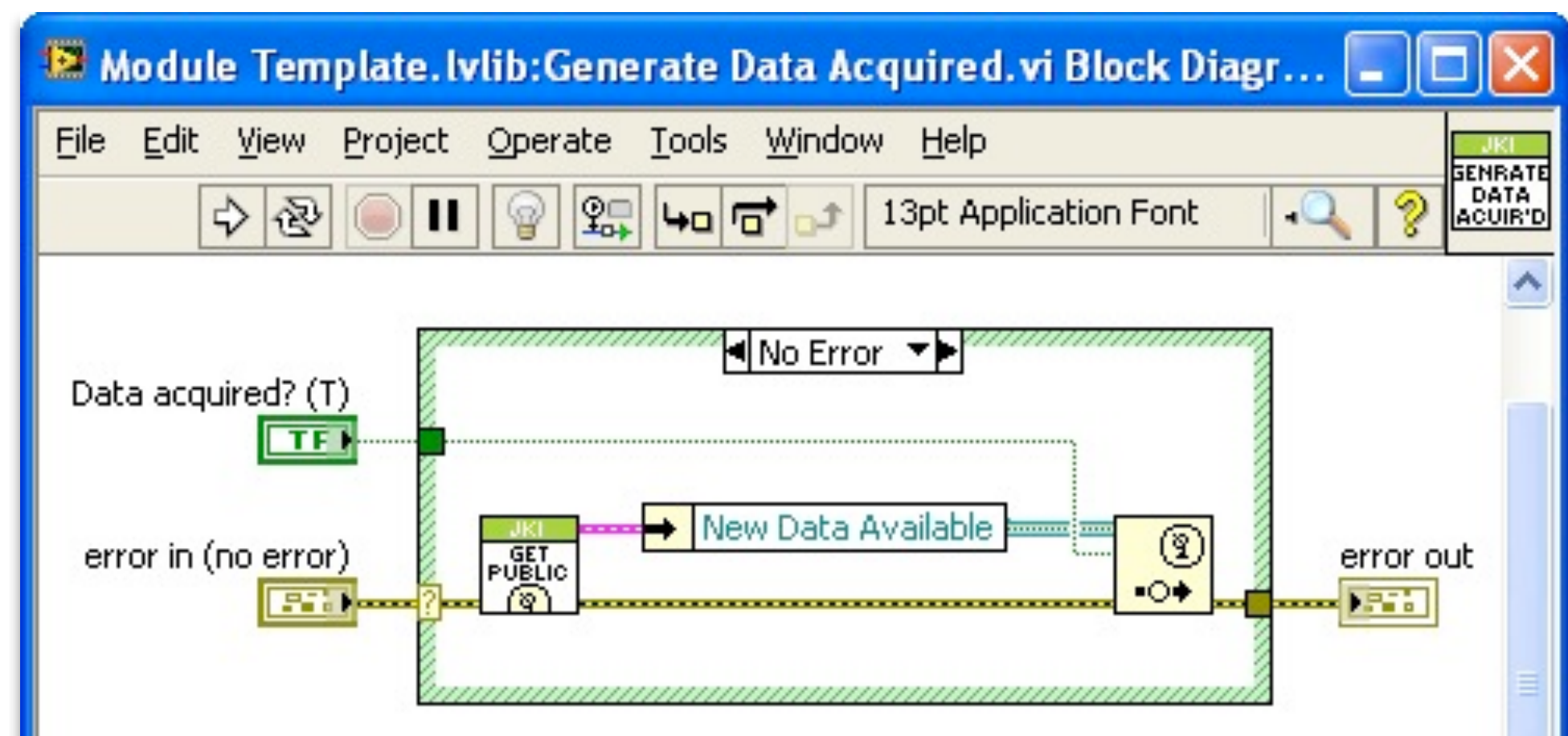
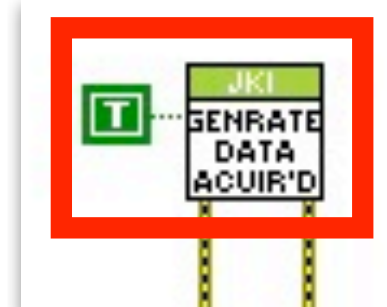
Process VI generates a Public Event...



# Public Events: Sending Information OUT



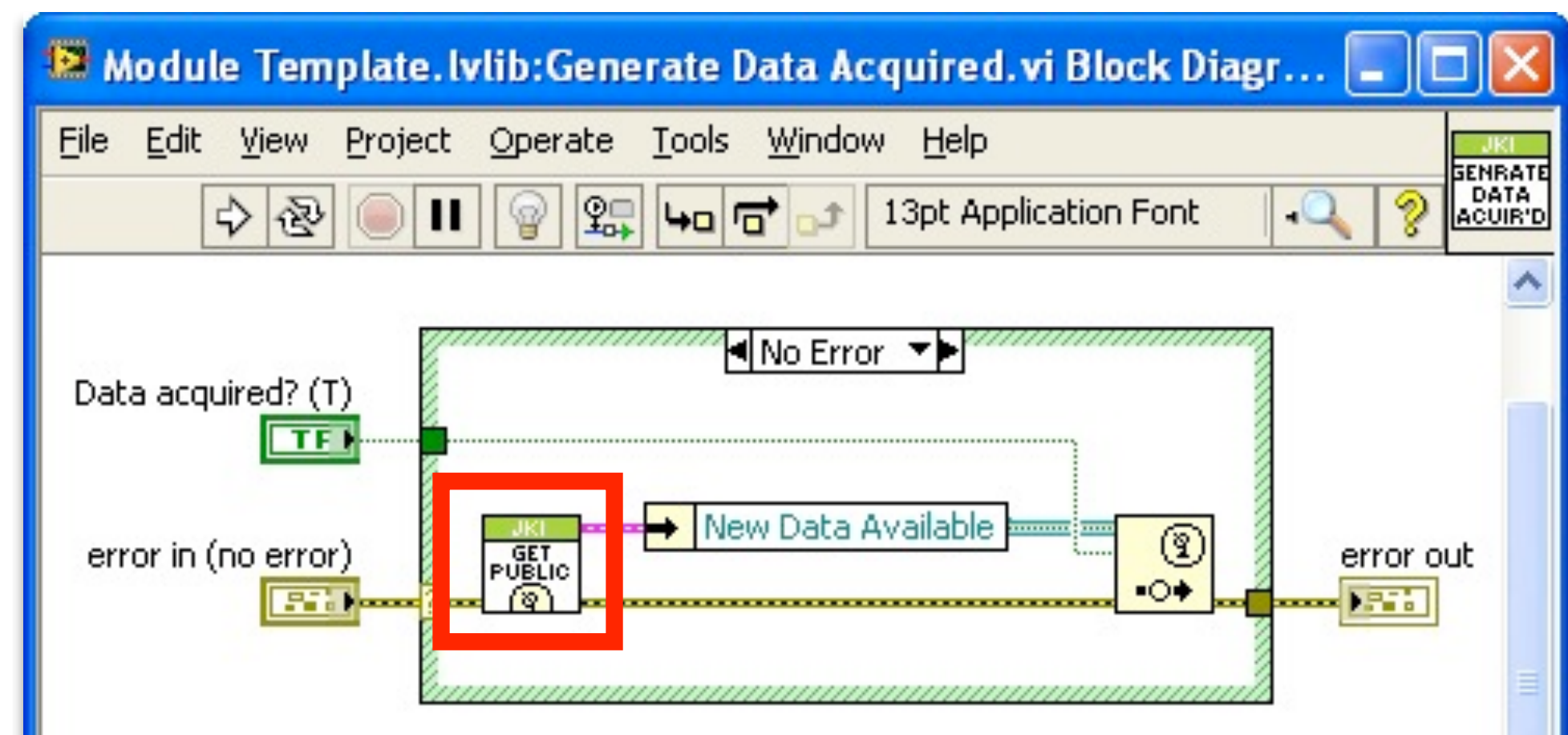
Process VI generates a Public Event...



# Public Events: Sending Information OUT



Process VI generates a Public Event...

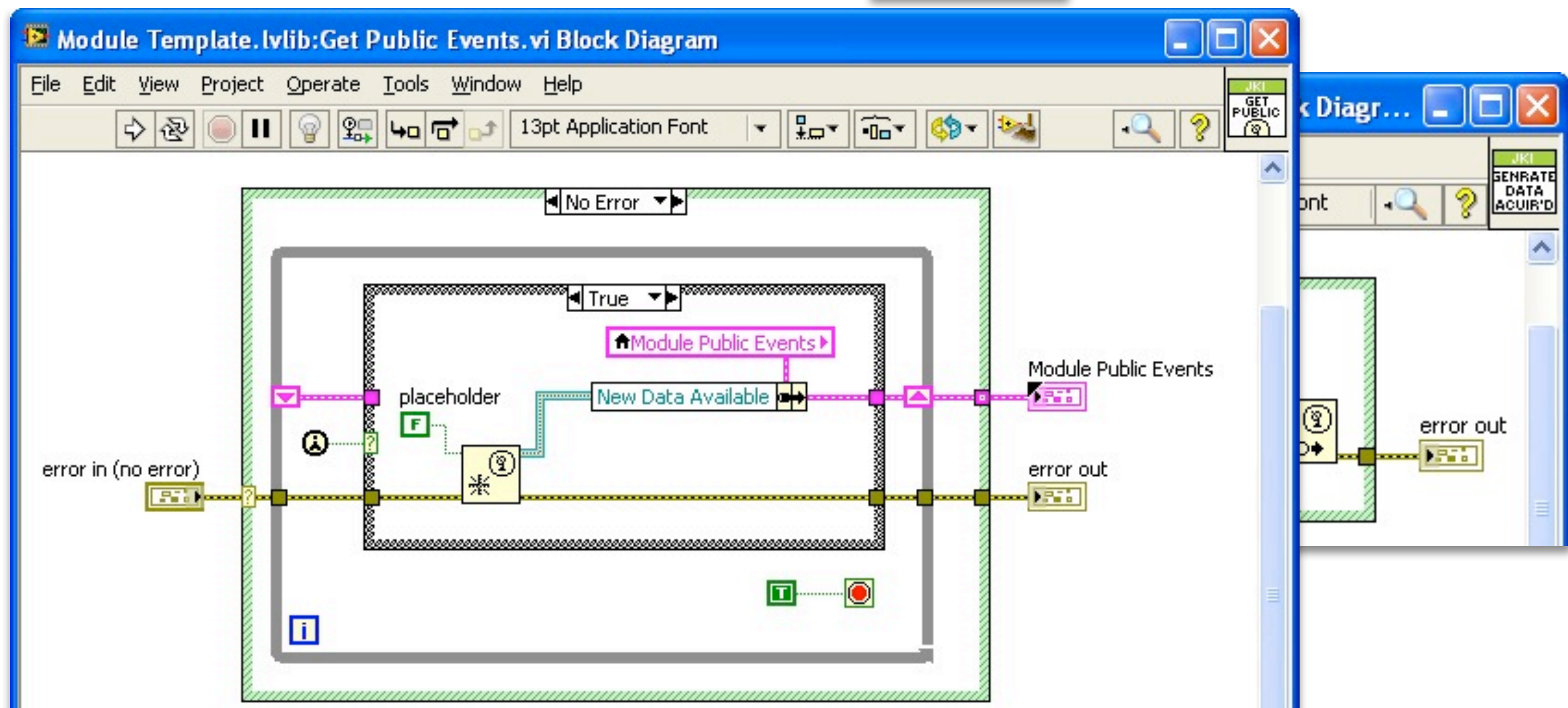




# Public Events: Sending Information OUT



Process VI generates a Public Event...





# Public Events: Sending Information OUT



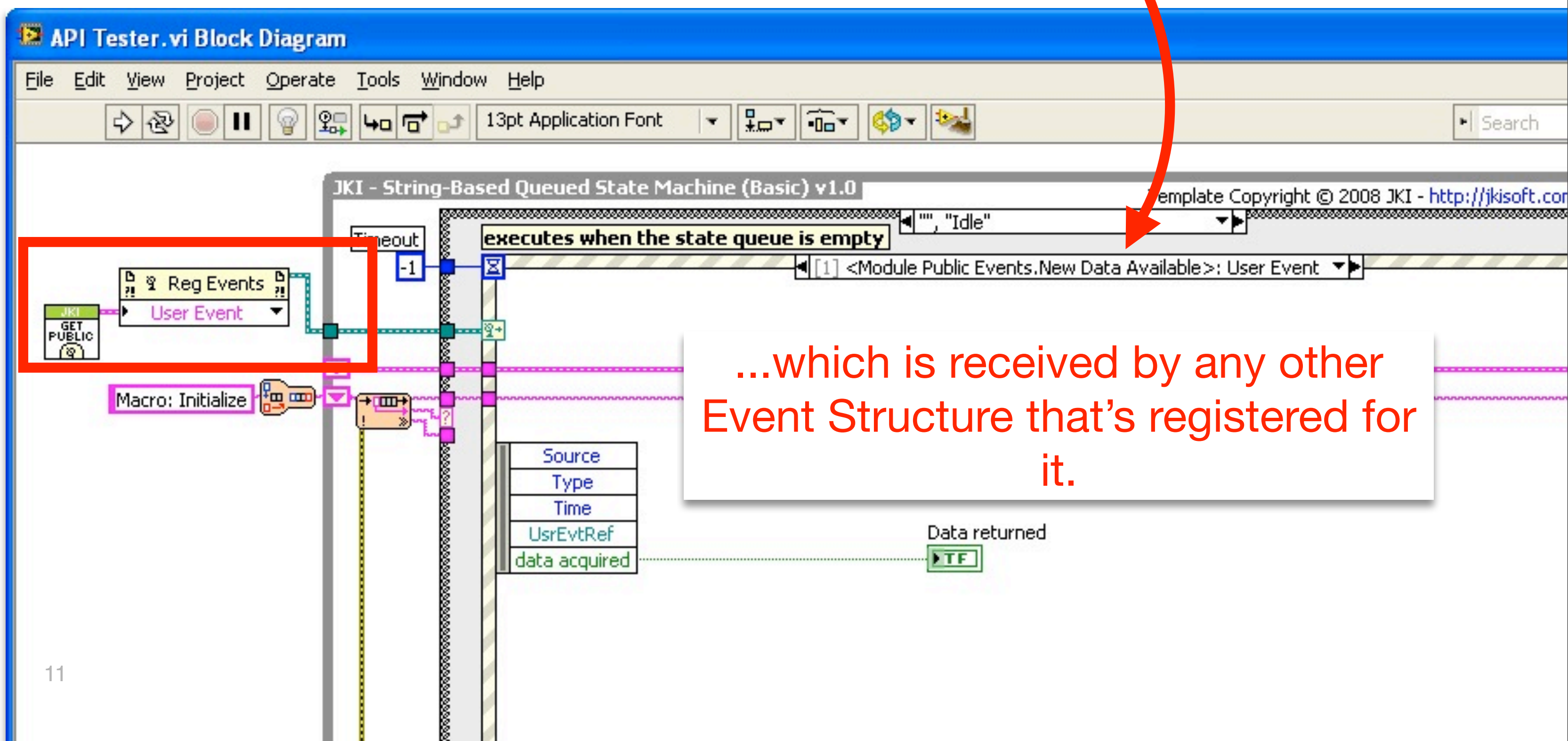
Process VI generates a Public Event...



# Public Events: Sending Information OUT



Process VI generates a Public Event...

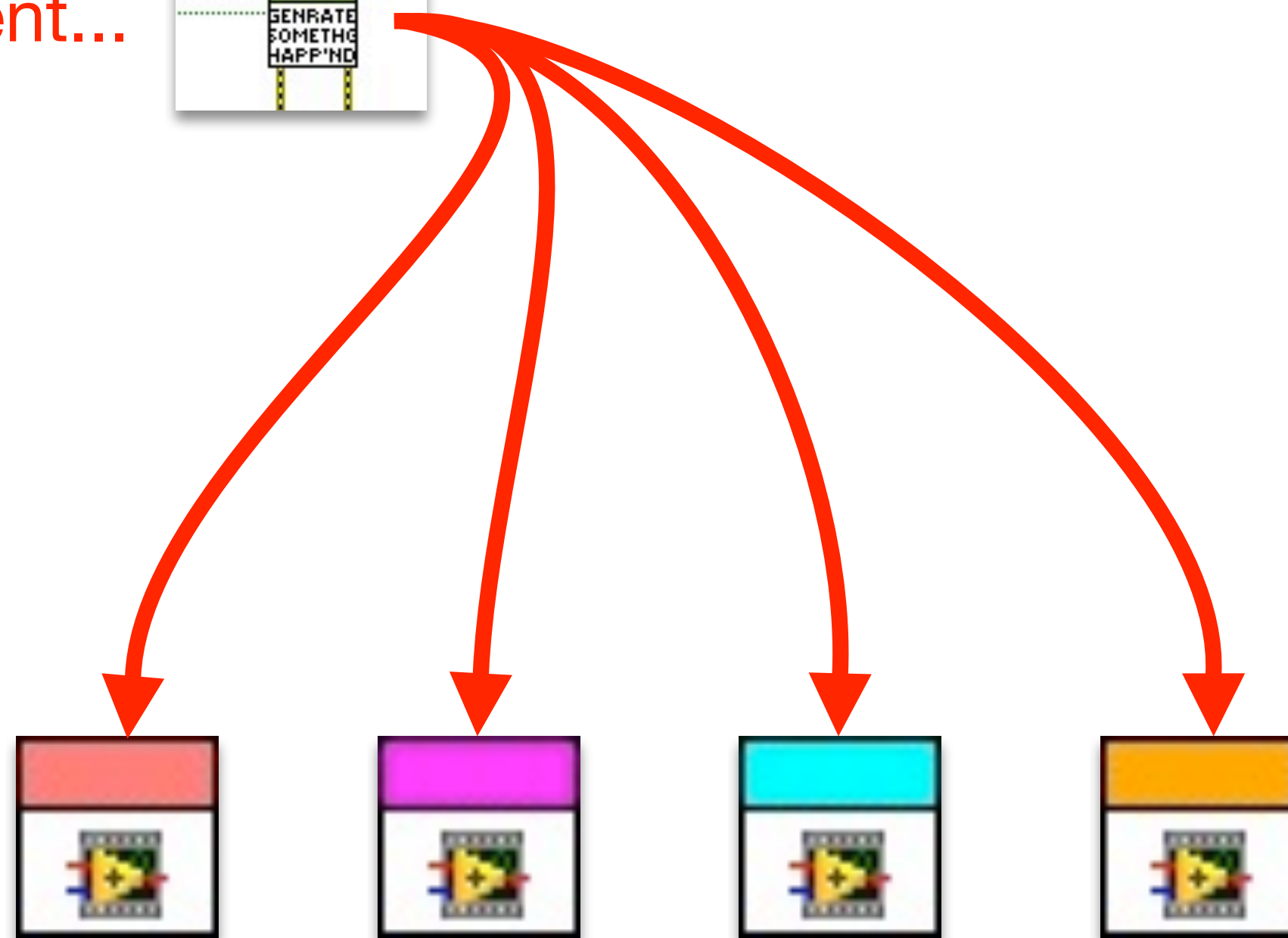


...which is received by any other Event Structure that's registered for it.

# Public Events: One to Many



Process VI generates a Public Event...



...that can be received by any number of consumers!

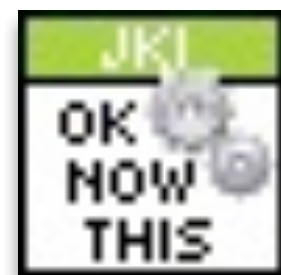
# The Public API



Public API VIs...



...generate Private Events...





# The Public API



Public API VIs...



...generate Private Events...

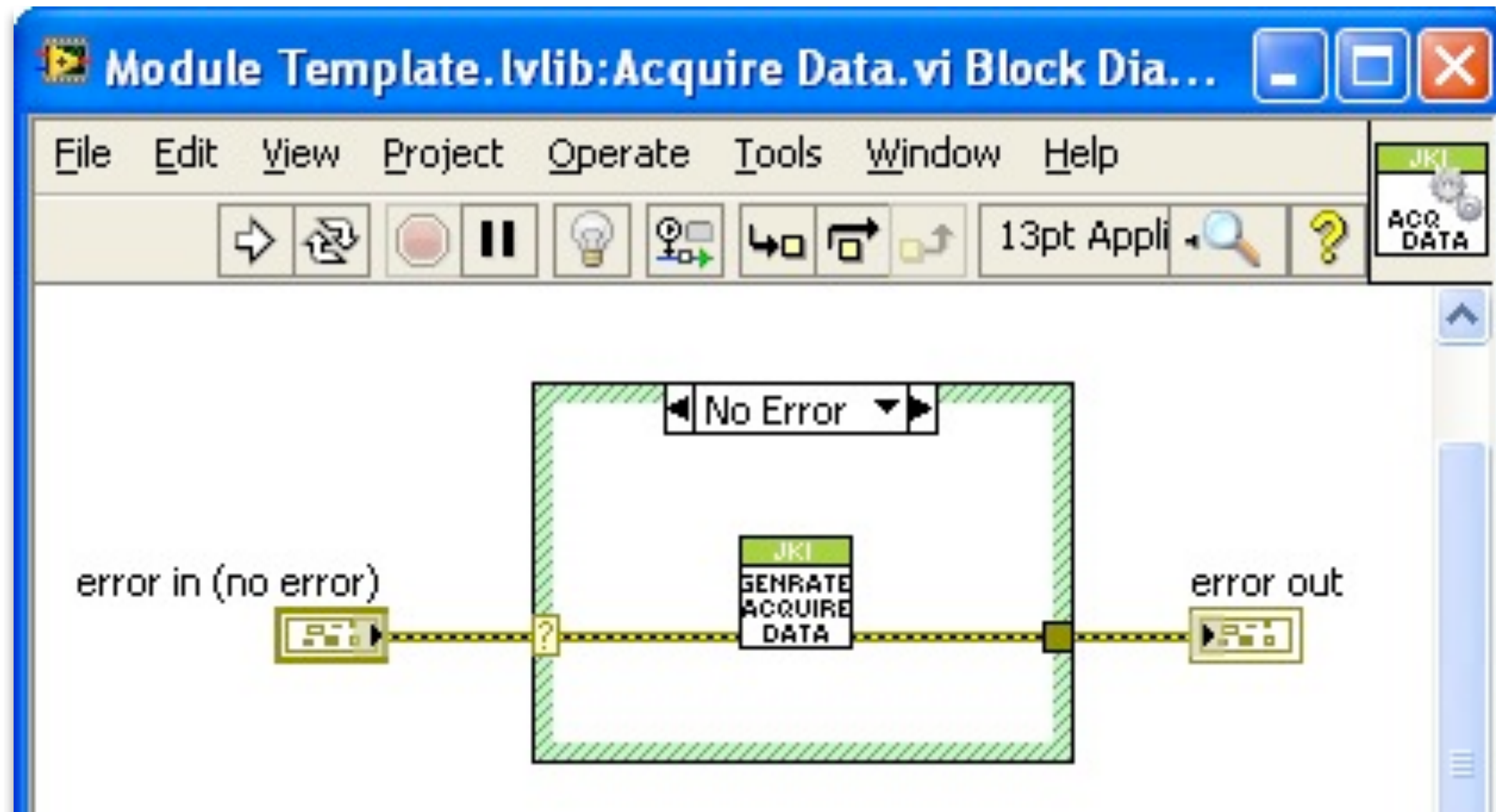


...that are handled by the asynchronous Process VI

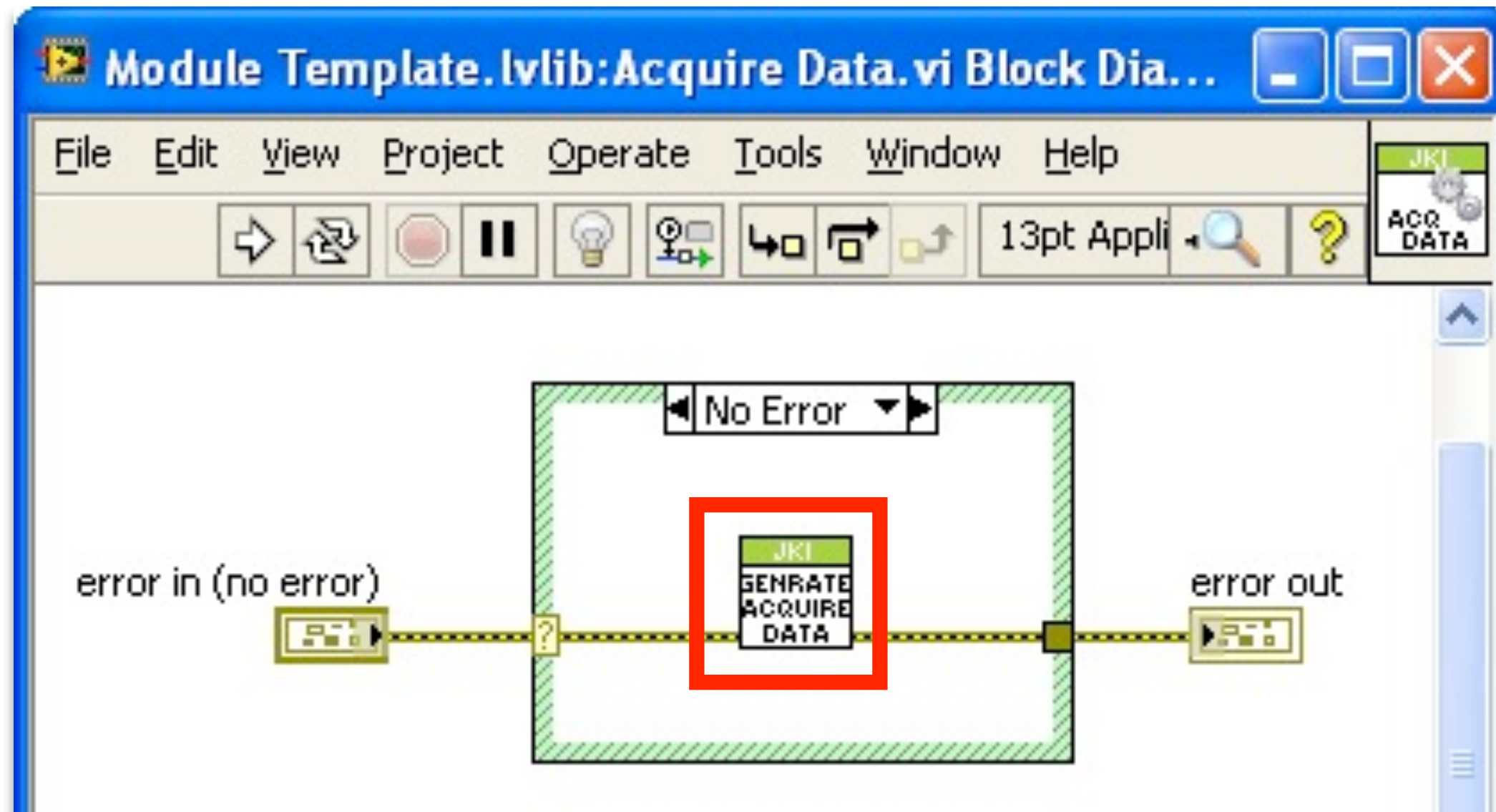
# Public API / Private Events



# Public API / Private Events

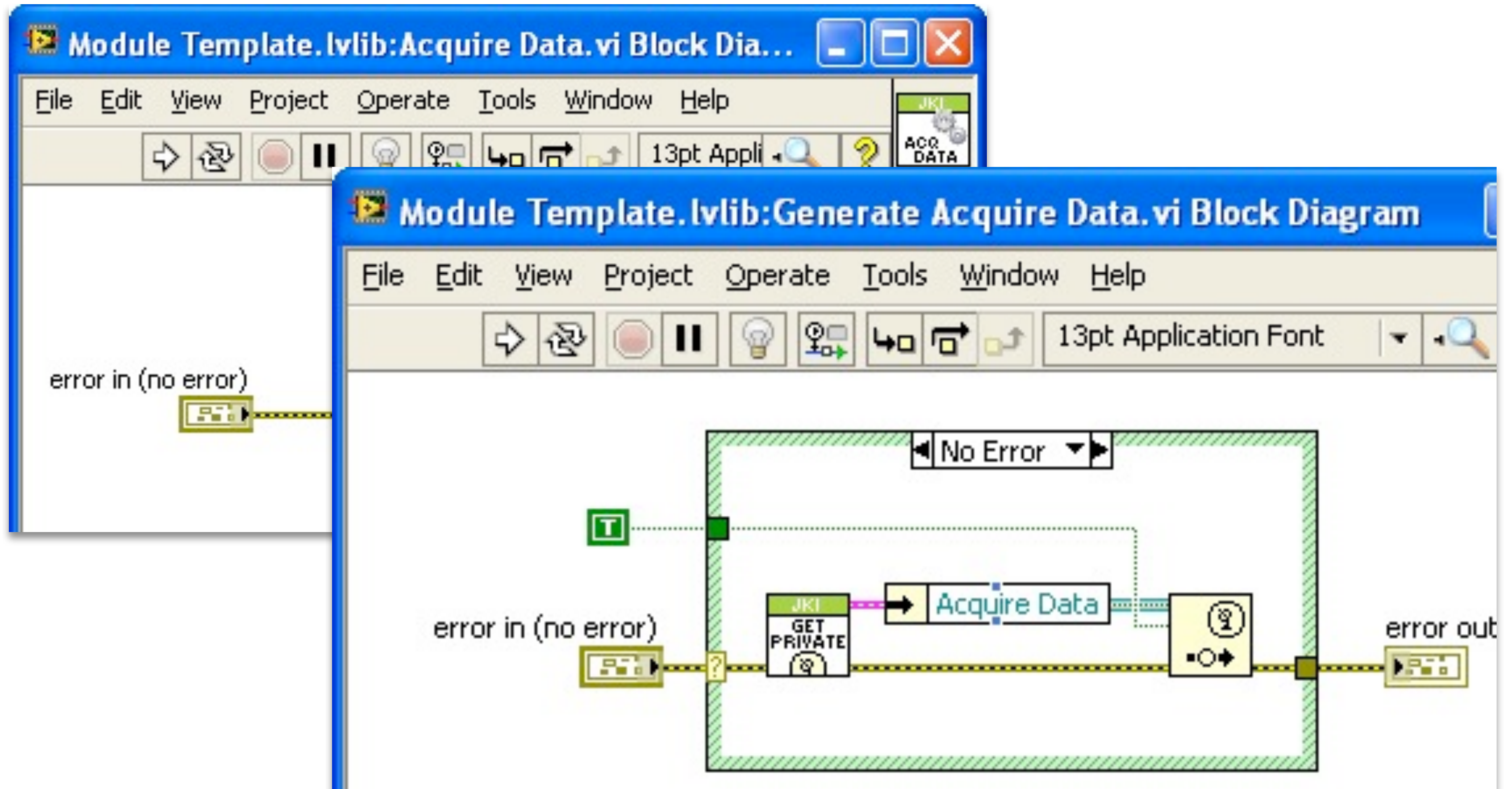


# Public API / Private Events



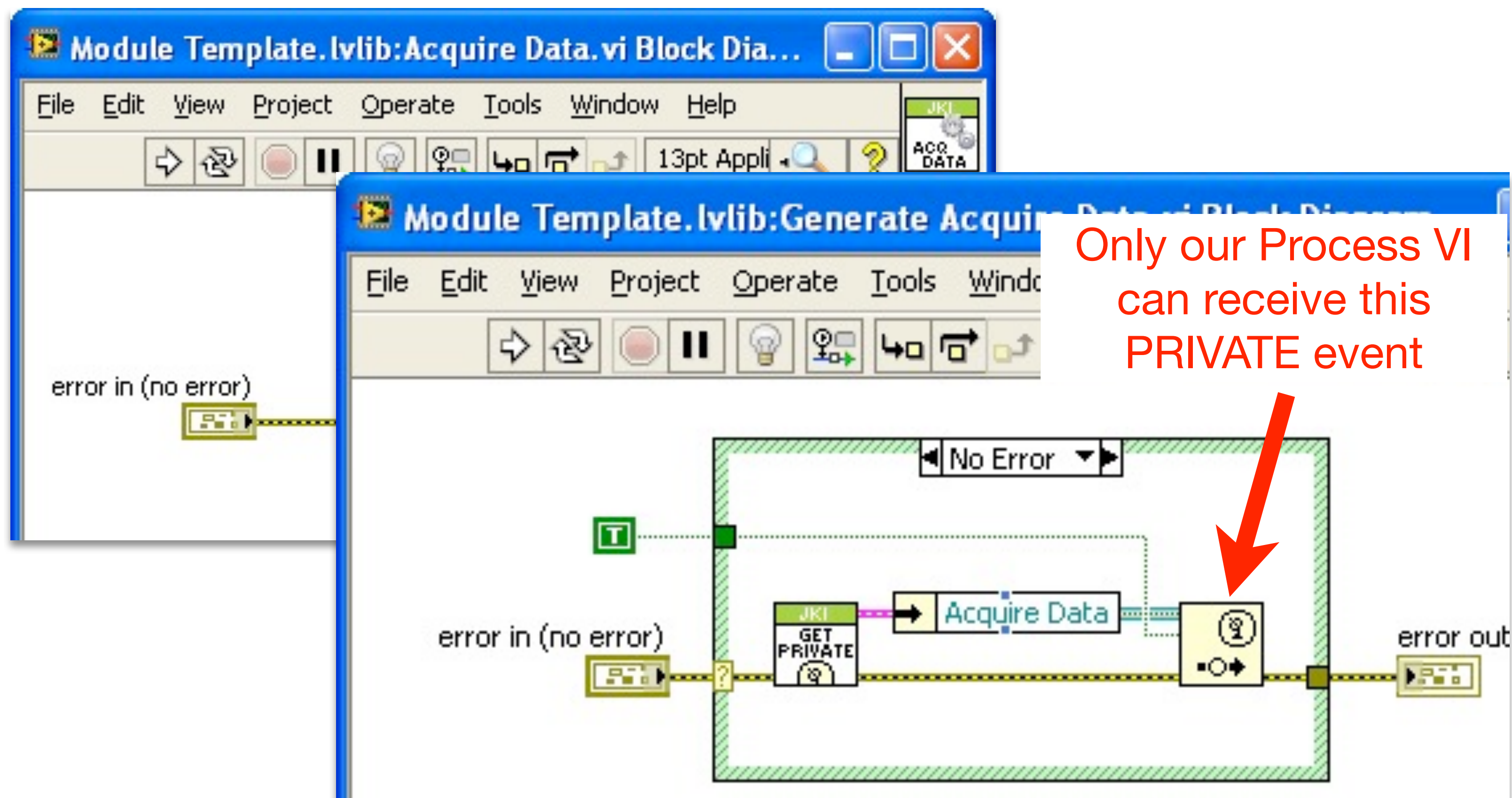


# Public API / Private Events

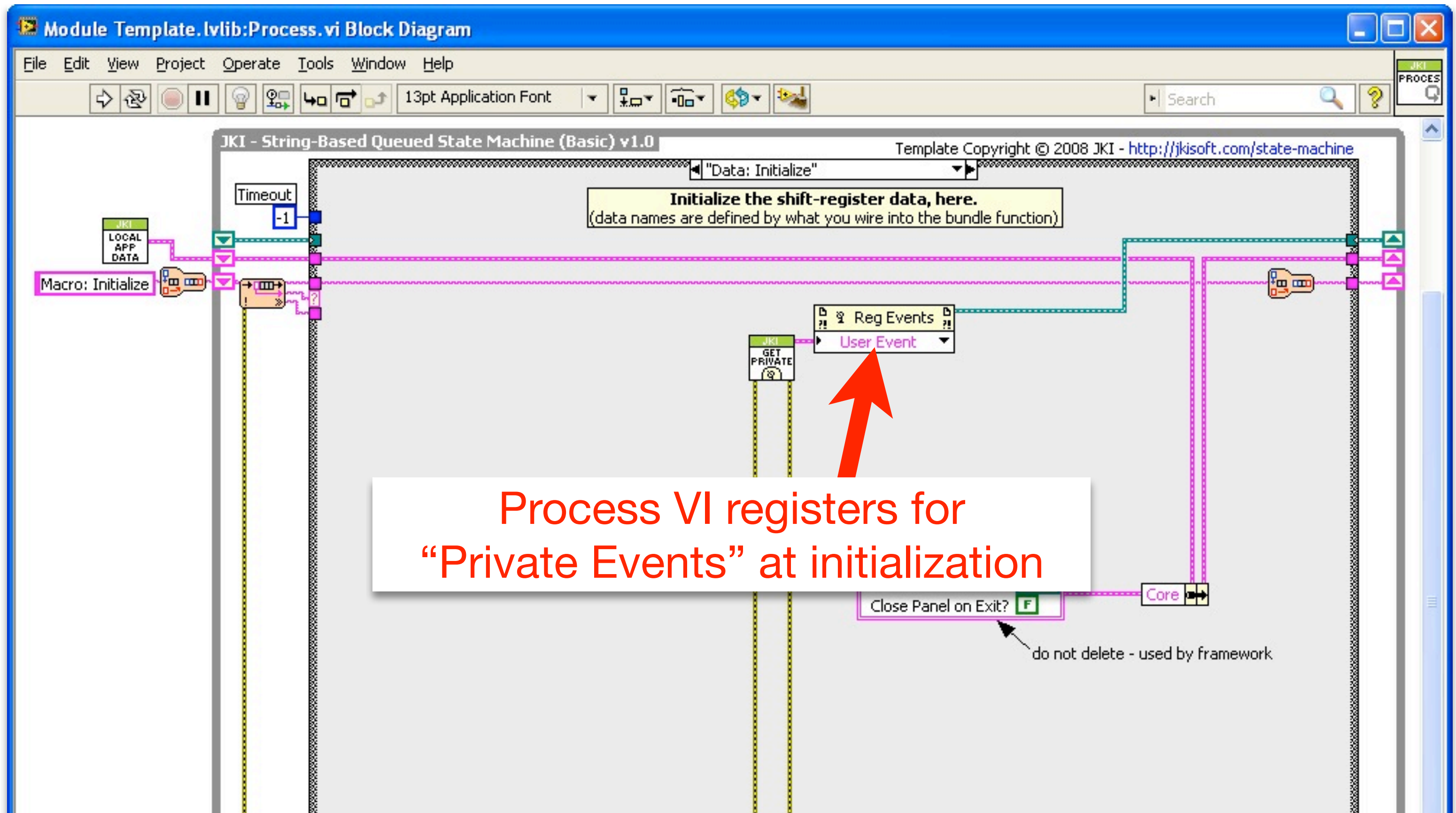




# Public API / Private Events

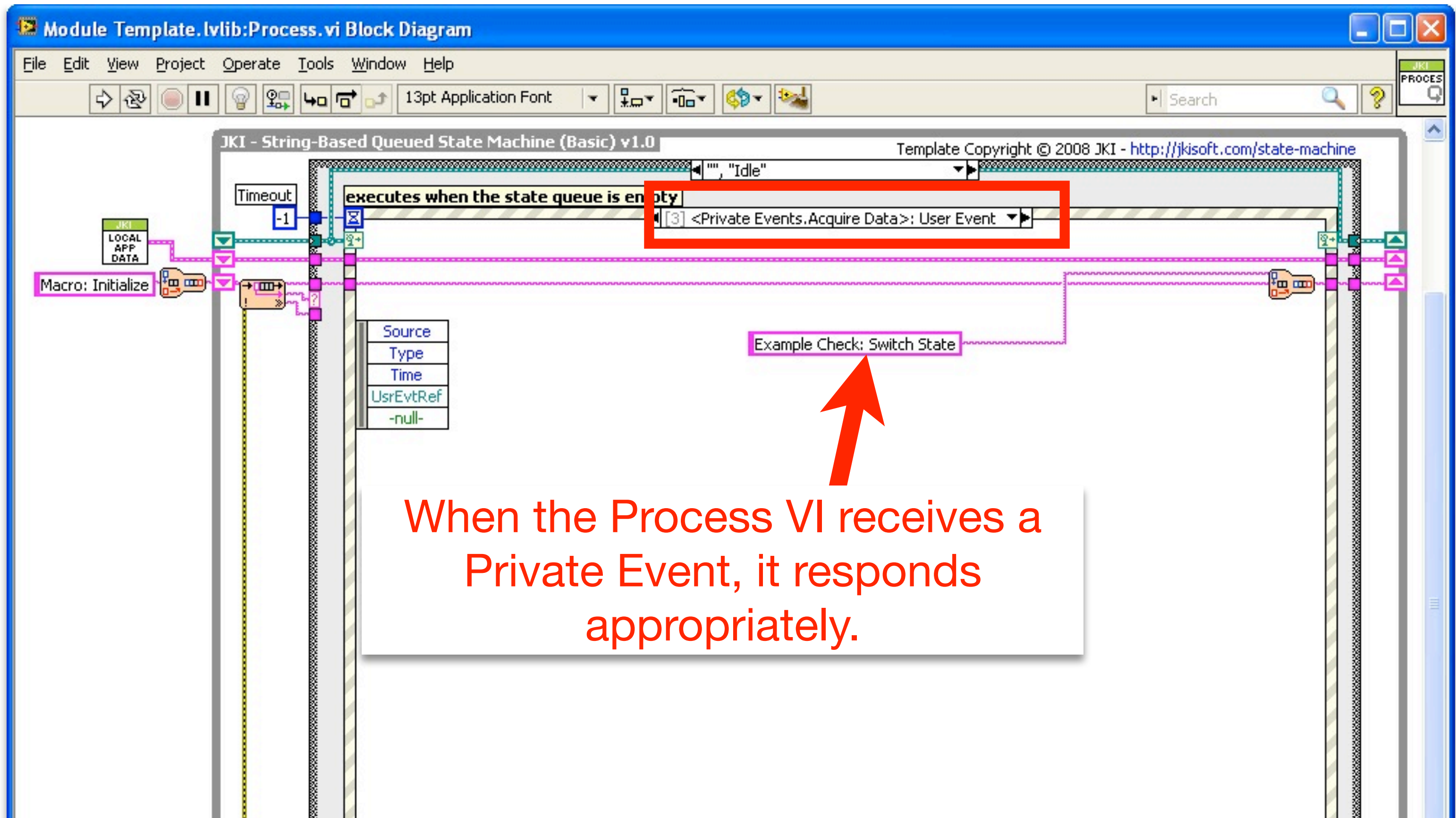


# Public API / Private Events





# Public API / Private Events

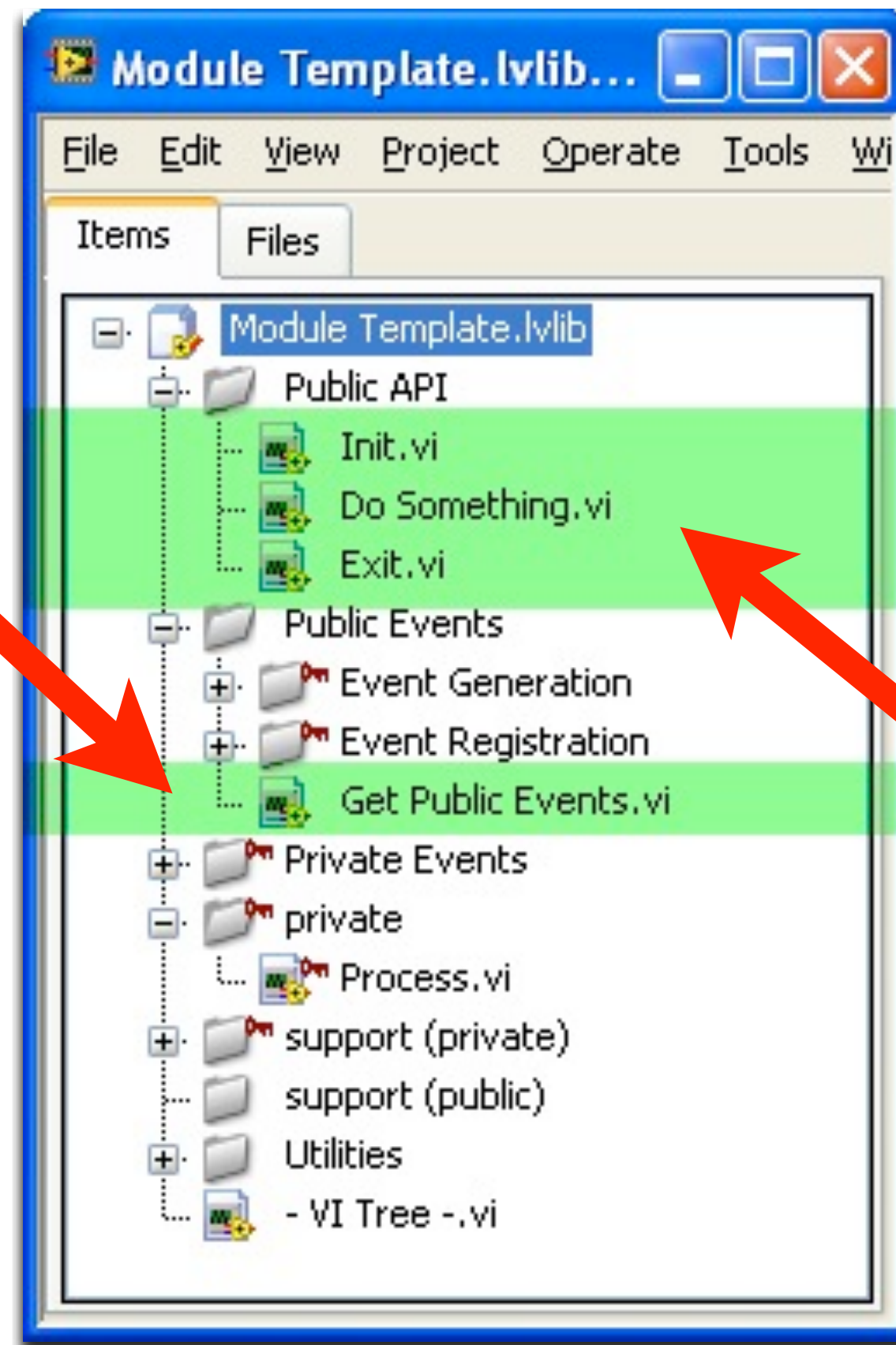




# Putting it All Together

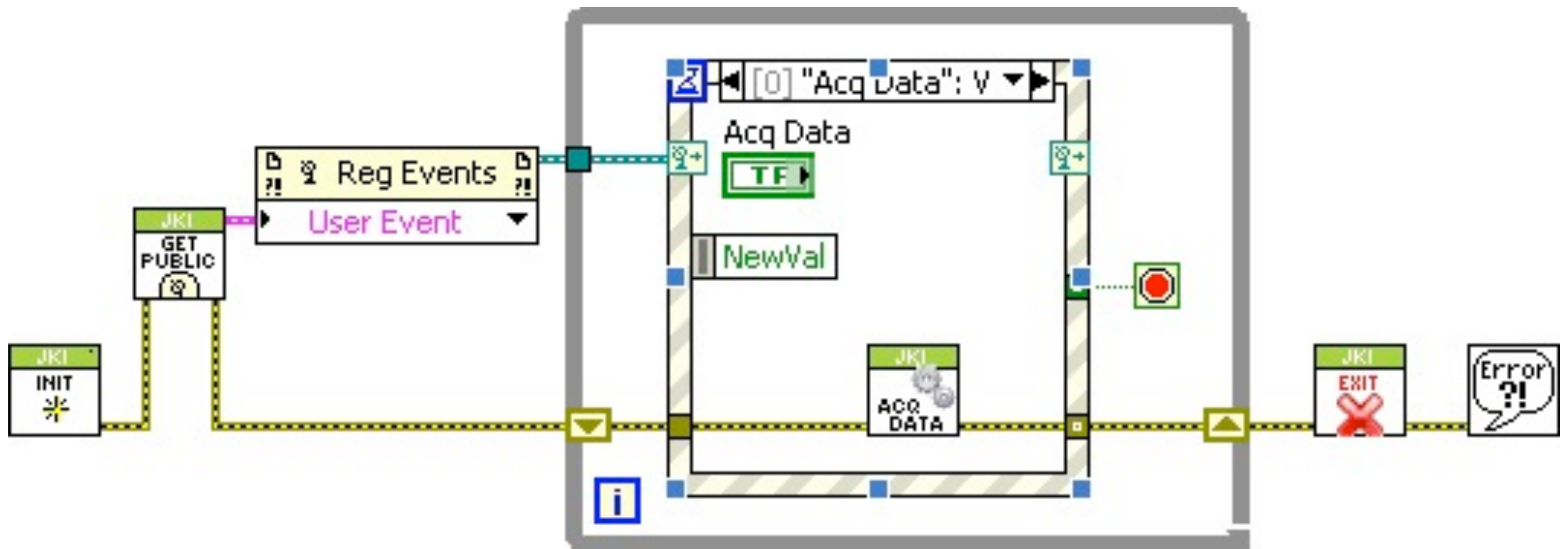


Client code  
registers for Public  
Events



Client code invokes  
Public API

# Putting it All Together

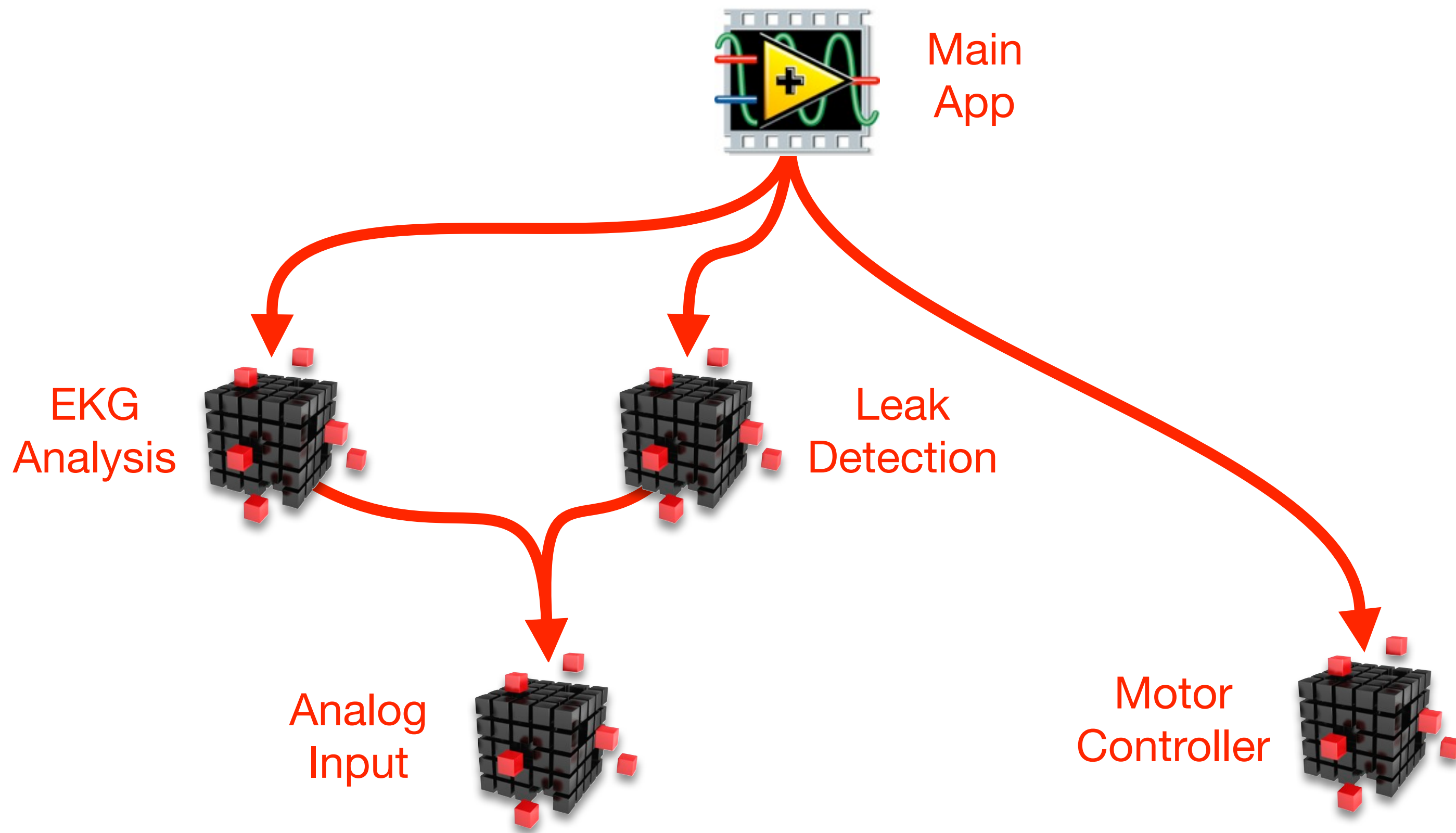
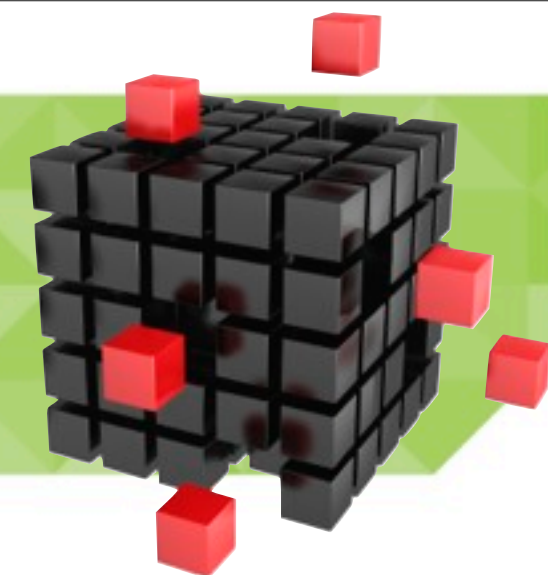


Start Process &  
register for Public  
Events

Invoke Public API  
as needed

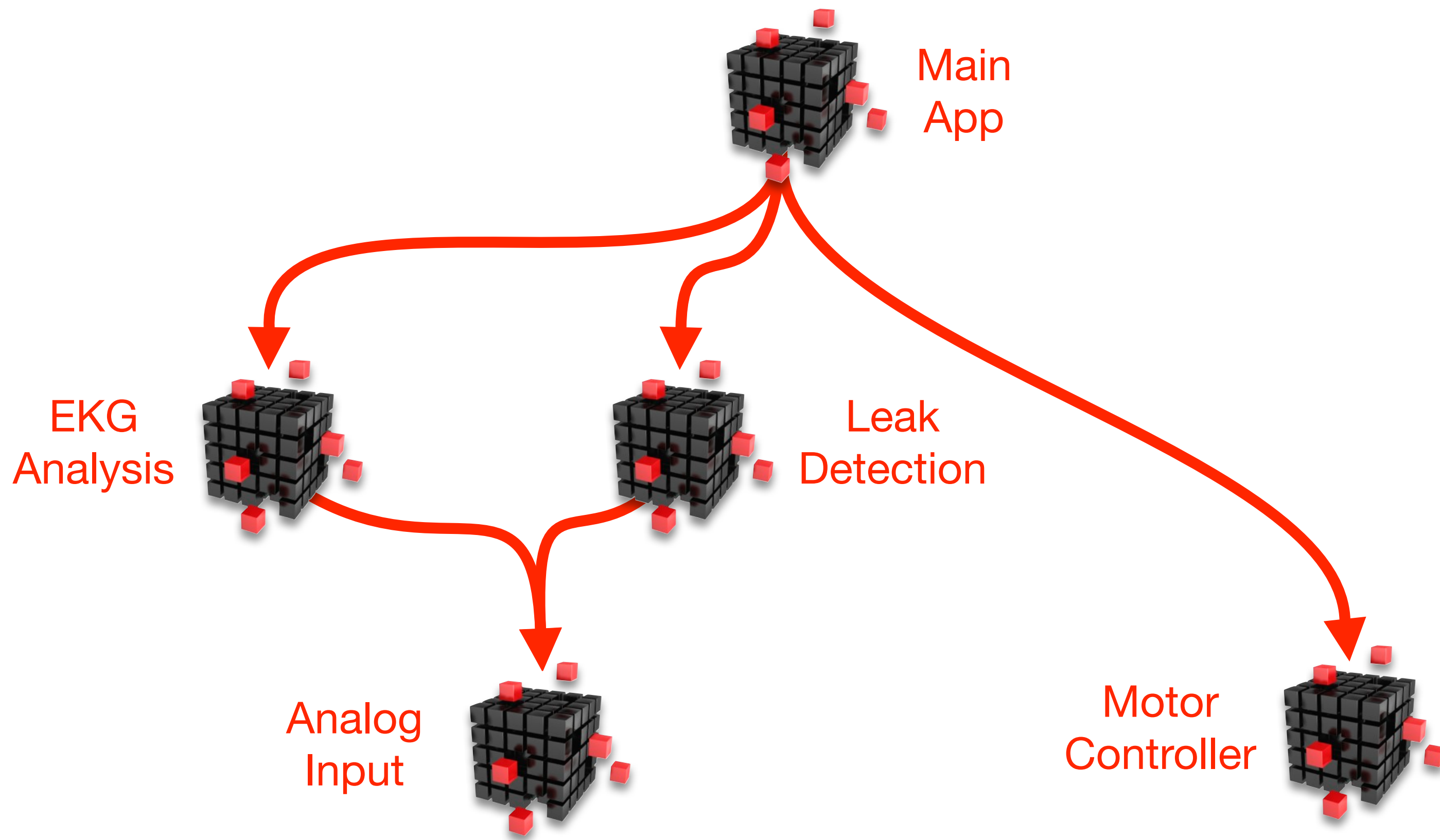
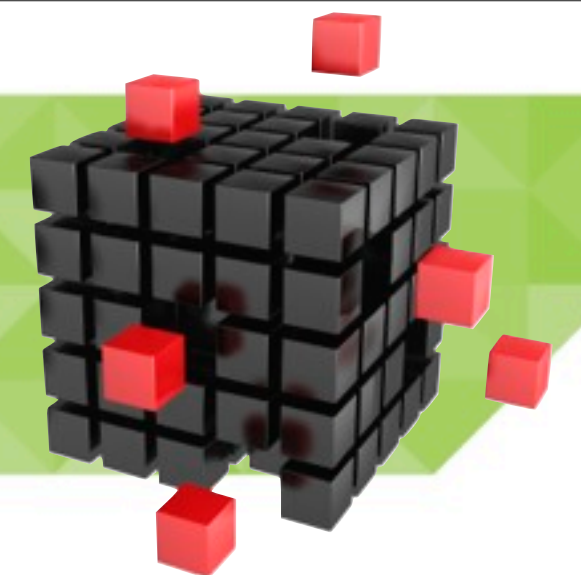
Exit Module  
when done

# Everything is a Module



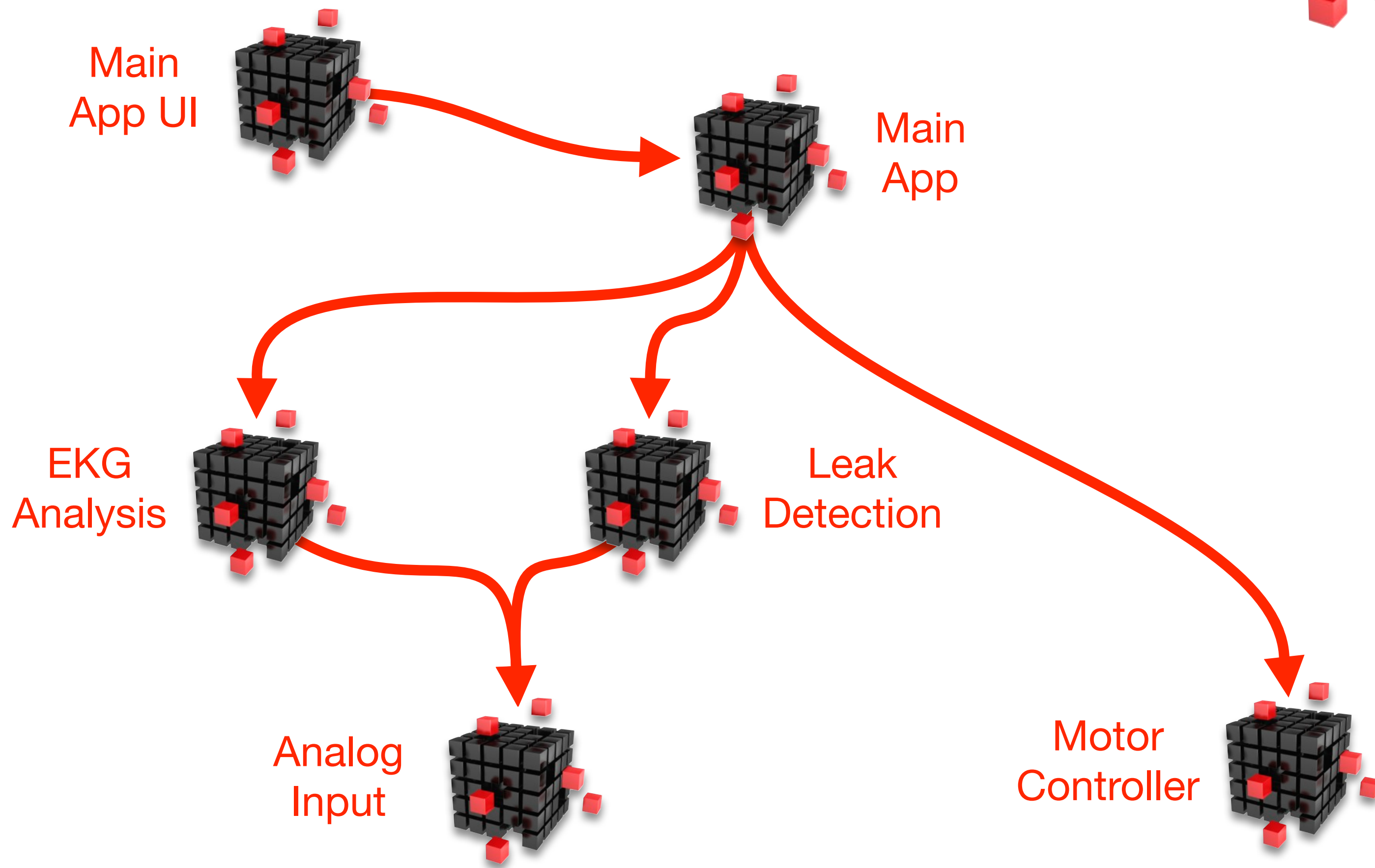


# Everything is a Module

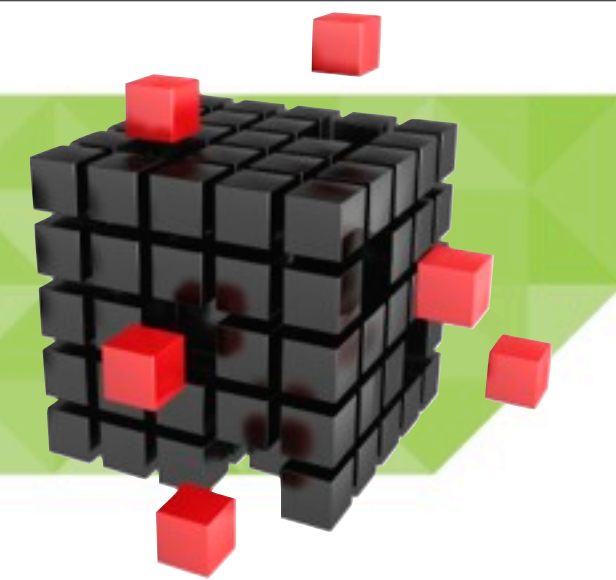




# Everything is a Module



# Extensions & Tweaks

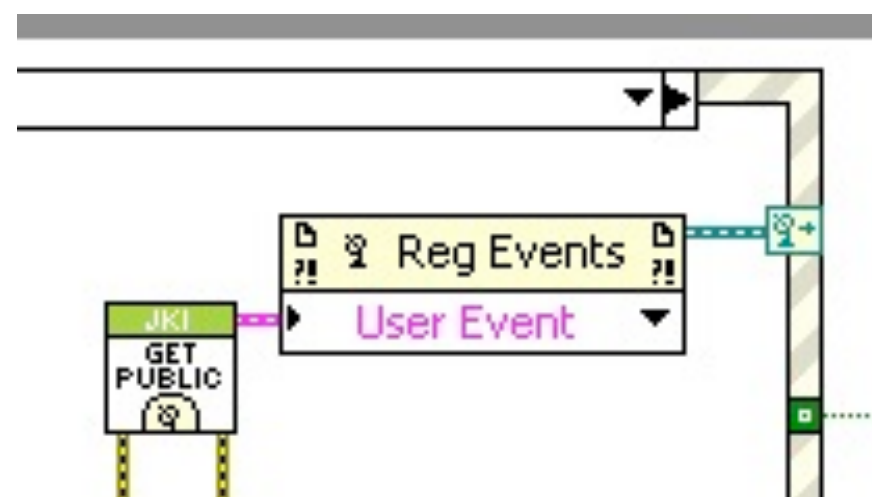


Use with by-value & by-ref objects

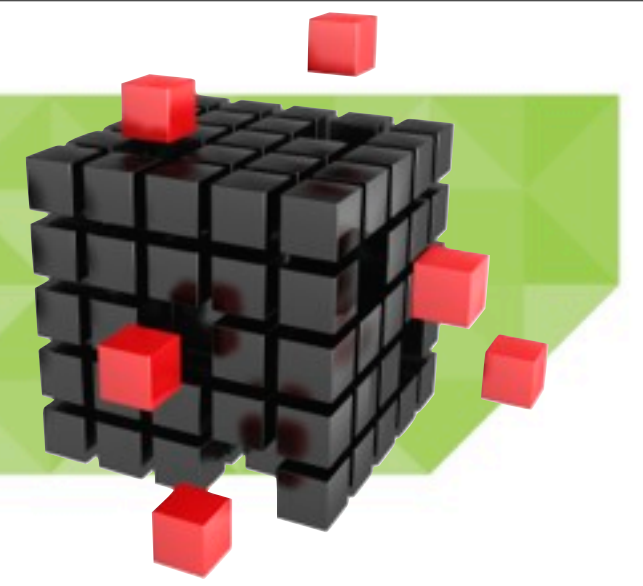
Community & Protected Events

Dynamically unregister / reregister

Synchronous communication



# Things That Suck



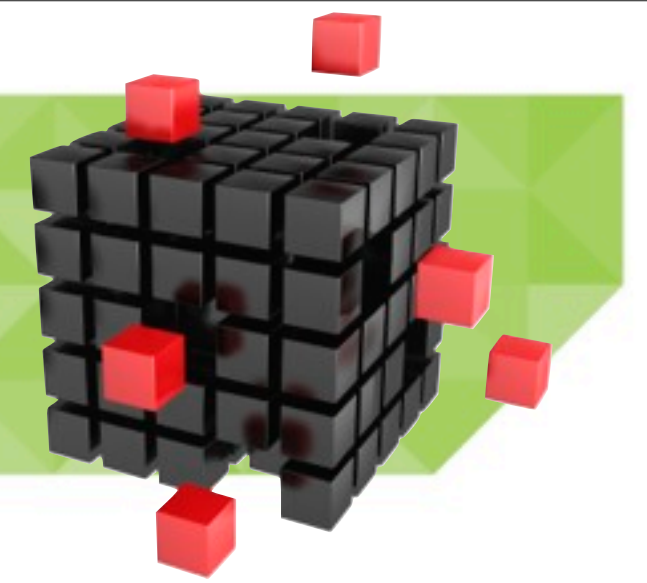
~~Most annoying bug ever.~~ Fixed in LV2011!

No Event queue management or introspection

No Notifier-like behavior (“ignore previous”)



# Take Home Point

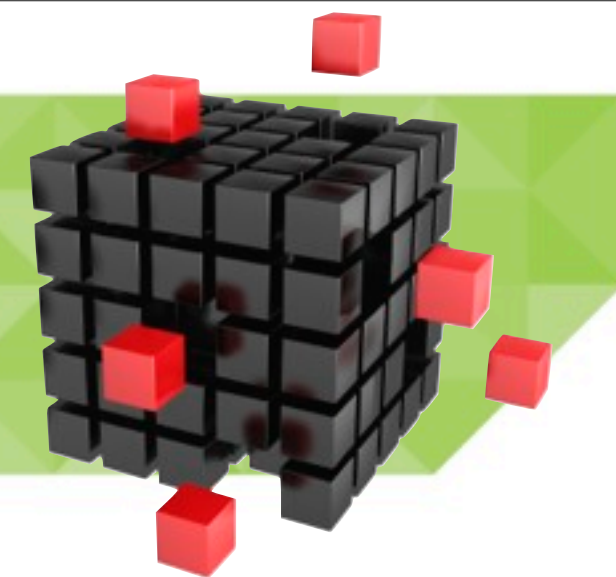


User Events are a very easy-to-use feature with a lot of cool functionality.

They form the basis of JKI's primary application frameworks & templates.

If we could get a couple things fixed/added to LabVIEW, we could do even better.

# The End!



## Discussion!