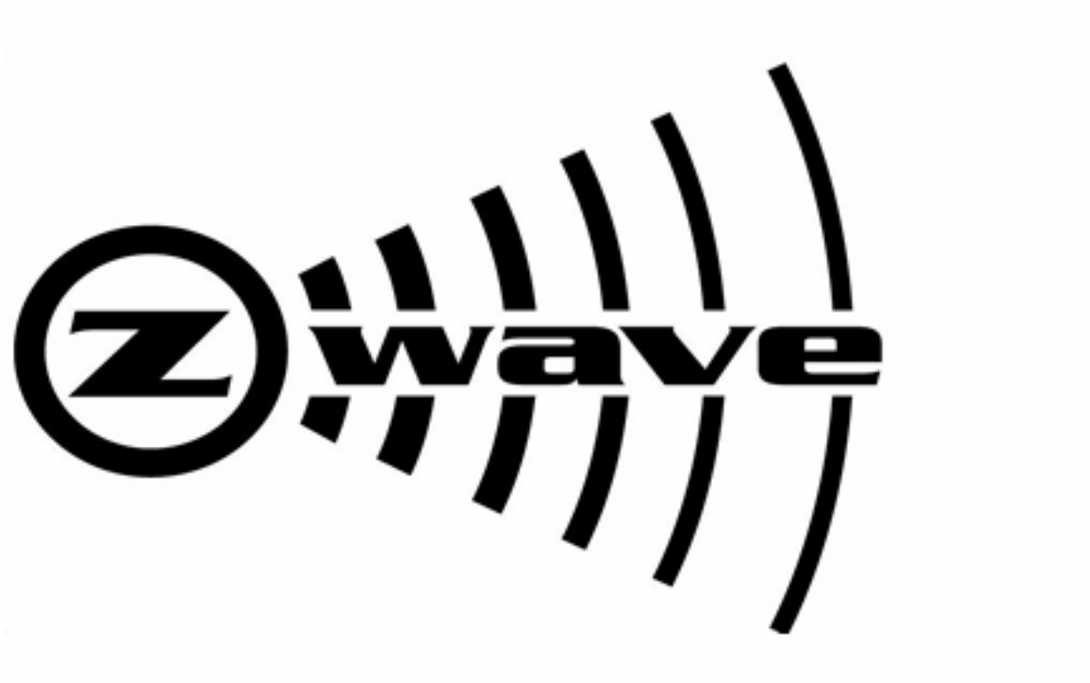
Home Automation

With Perl

Home Automation

With very little Perl


Z
wave

Standardised (up to the Application Layer, though not open) • Multiple Vendors • Interoperable, Secure, Routable • Long lasting battery powered devices • Affordable (not cheap) • Many different device types

Primary controller
usually IP gateway

Secondary Controller

usually remotes

Sensors & Actuators

lights, climate, windows, security, …

Mains vs. battery powered devices

Associations vs. Scenes

RaZberry

http://razberry.zwave.me

Z-Way

http://en.z-wave.me

Homely

https://github.com/maros/Homely

Single Process Event Loop

AnyEvent, Twiggy, Mojolicious

Initialiser

Core

AnyEvent loop

Web

Components

State
(eg. Weather)

Scene
(eg. Night)

Connectors

Z-Wave

Singletons (DI planed)

Auto discovery of components (M::Pluggable)

State has lifecycle

Roles: Ephermal, Permanent, Temporary

States serialised with MooseX::Storage

Connector uses Inline::C

API trouble

Sparse Documentation

Only one instance possible

Threads

Inline-C swallows warnings

```
use Inline
C               => 'Config',
BUILD_NOISY     => 1;
```

```
int fd[2];

int myzway_init(int loglevel) {
        pipe(fd);
        ...
        return fd[0];
}
```

```
sub init {
    my ($self) = @_;
    my $fh = IO::Handle->new();
    my $fd = myzway_init(2);
    if ($fd) {
        $fh->fdopen($fd,'r');
        $fh->blocking(1); # ?
    }
    my $io = AnyEvent->io(fh => $fh, poll => "r", cb => sub {
        my $input = $fh->getline;
        return
            unless defined $input; # !!!
        ...
    });
    $self->io($io); # keep ref to $io
    return $self;
}
```

~~Winter~~ Summer is coming

Vera

https://getvera.com

OpenWRT & MIOS

http://mios.com

LUA scripting engine

DEMO

```
function device_on(device)
   device = tonumber(device)
   luup.log("[MyHome] Turn device on " .. device)
   luup.call_action("urn:upnp-org:serviceId:SwitchPower1",
      "SetTarget", { newTargetValue = 1 }, device)
end
```

```lua
function device_move(device,percentage)
	device 			= tonumber(device)
	percentage 		= math.floor(tonumber(percentage))
	local related 	= device_attr(device,"related")
	local class 	= device_attr(device,"class")

	if percentage == nil then
		percentage = BLINDS.PARTIAL
	end

	-- Check related devices
	if related ~= nil then
		local related_position = device_position_get(related)
		if class == "Blind" and percentage < 60 and related_position > 40 then
			percentage = 60 -- lower is further
		elseif class == "Window" and percentage > 40 and related_position < 60 then
			percentage = 40 -- higher is further
		end
	end

	luup.log("[MyHome] Moving device "..device.." to "..percentage.."%")

	if (percentage == 100 or percentage == 0) then
		device_position_set(device,percentage)
	else
		local device_position  = device_position_get(device)
		local device_timer 		= device_attr(device,"timer")
		local device_diff 		= device_position - percentage
		local device_time 		= math.floor(device_diff / 100 * device_timer)
		local device_direction = "down"
		if device_time < 0 then
			device_time = device_time * -1
			device_diff = device_diff * -1
			device_direction = "up"
		end

		if device_diff >= 15 and device_time > 2 then
			--percentage 				= math.floor(( 1 -( device_time / device_timer)) * 100)
			device_position_set(device,percentage)
			luup.log("[MyHome] Moving device "..device_time.."s to reach "..percentage.."%")

			local delay = math.random(10,20)
			luup.call_action(SID_WINDOWCOVERING, "Stop", {}, device)
			luup.call_delay("device_"..device_direction, delay, tostring(device))
			luup.call_delay("device_stop", device_time+delay, tostring(device))
		end
	end
end
```

1800++ lines of madness

HomelyAlarm

https://github.com/maros/HomelyAlarm

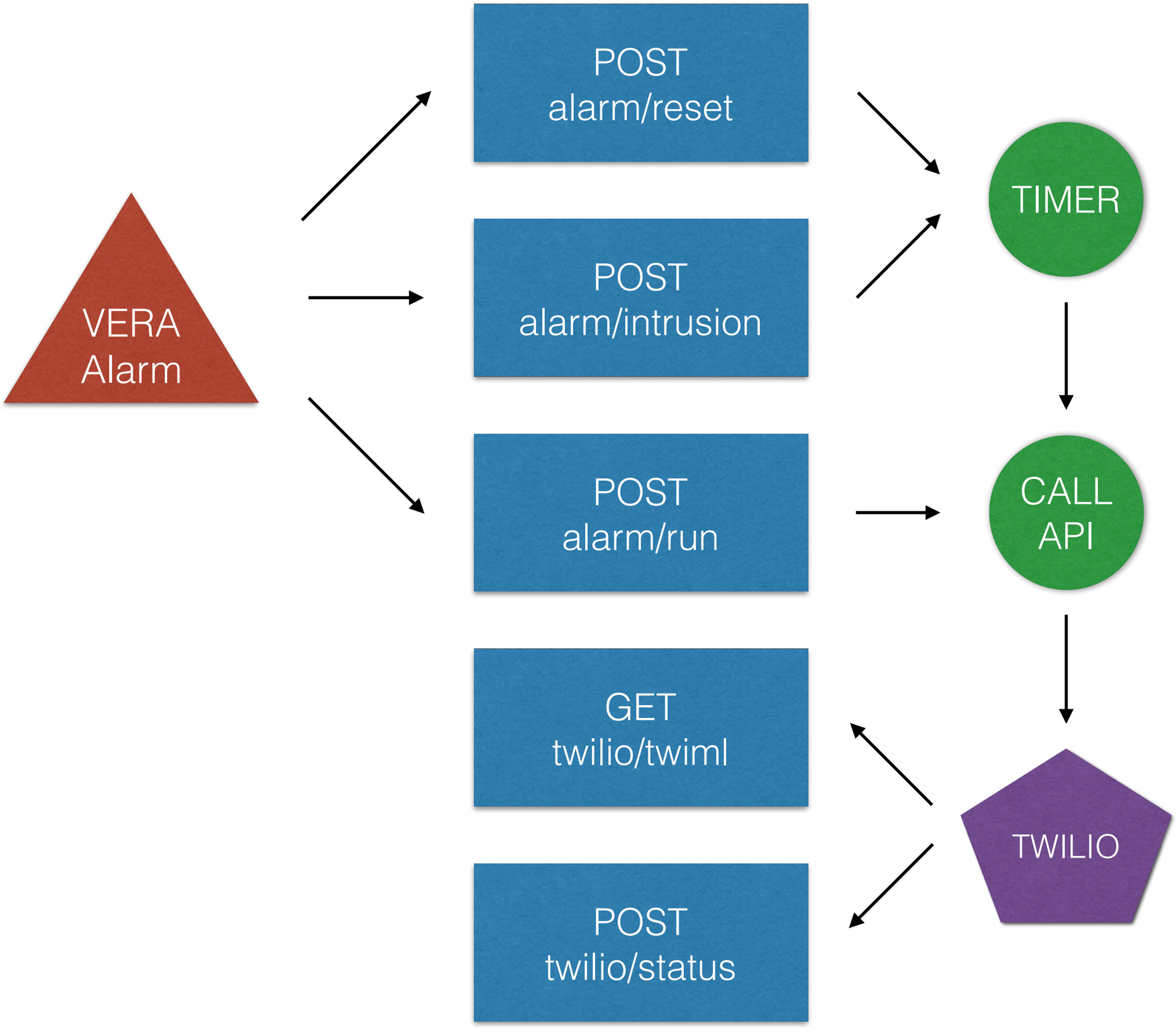
PSGI

(no additional web framework)

AnyEvent, Twiggy, MooseX::App

Twilio API via AnyEvent::HTTP

CLI for managing alarm message recipients


VERA
Alarm
POST
alarm/reset
POST
alarm/intrusion
POST
alarm/run
GET
twilio/twiml
POST
twilio/status
TIMER
CALL
API
TWILIO

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <Gather timeout="10" finishOnKey="*">
        <Say>Please enter your pin number and then press star.</Say>
    </Gather>
</Response>
```