

Extend Zabbix Agent 2 with Your Plugin

Ēriks Šneiders, Senior Software Developer, Zabbix



Welcome!

A brief intro about me and Zabbix agent 2 loadable plugins.

Introduction

Who am I ?

A Senior Software Developer, working mostly with Zabbix agent 2 and plugins!

What will we do today?

Learn to use loadable plugin and learn what to do to make it your own!

How will we do it today?

Go through an Example plugin and get technical with code!

Plugin Architecture

How does it work?

Core concepts

Separate repository

They are kept in Zabbix git but are not part of the main Zabbix repository.

Configuration

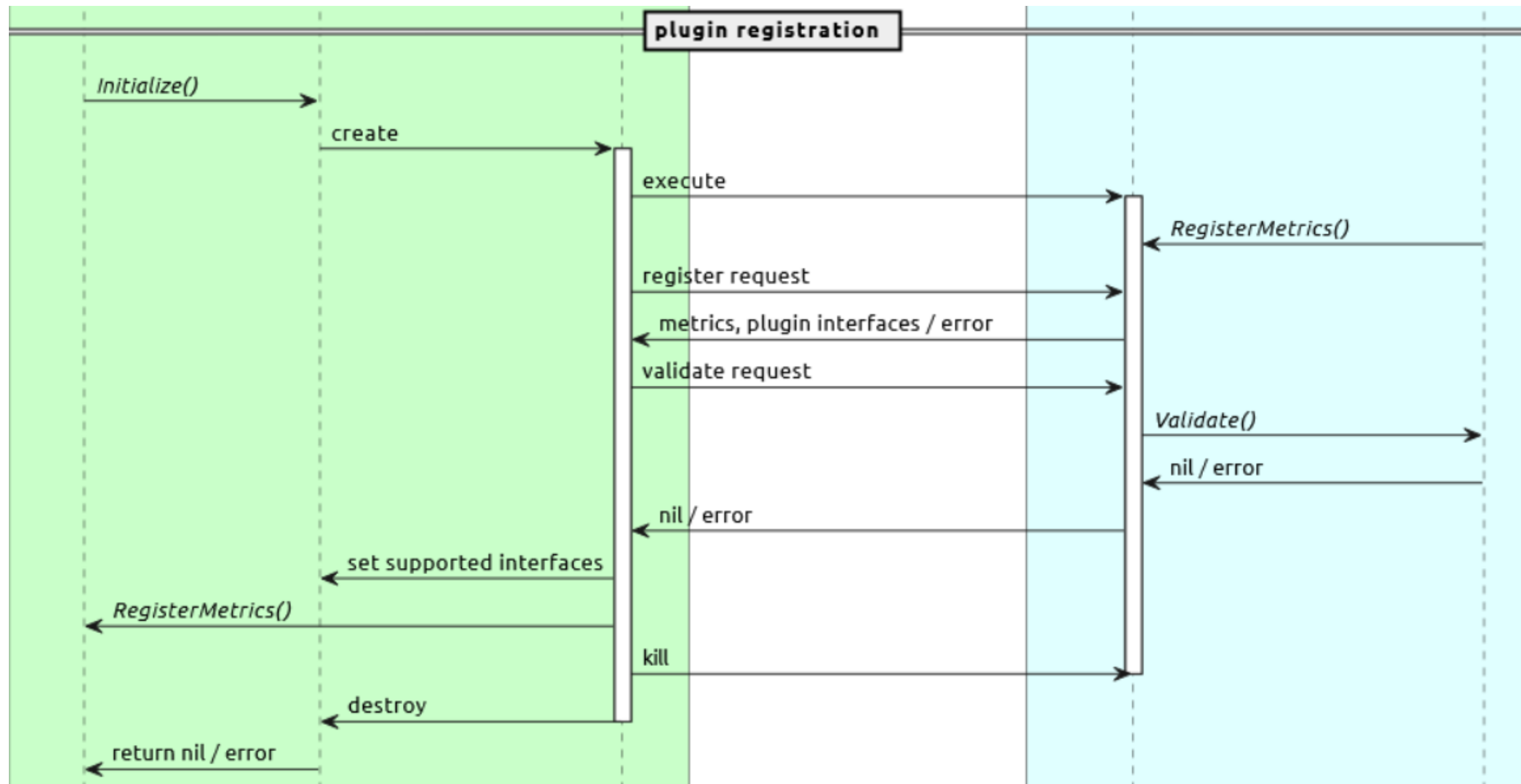
They have separate configuration files but are part of the Zabbix agent 2 configuration.

Communication flow

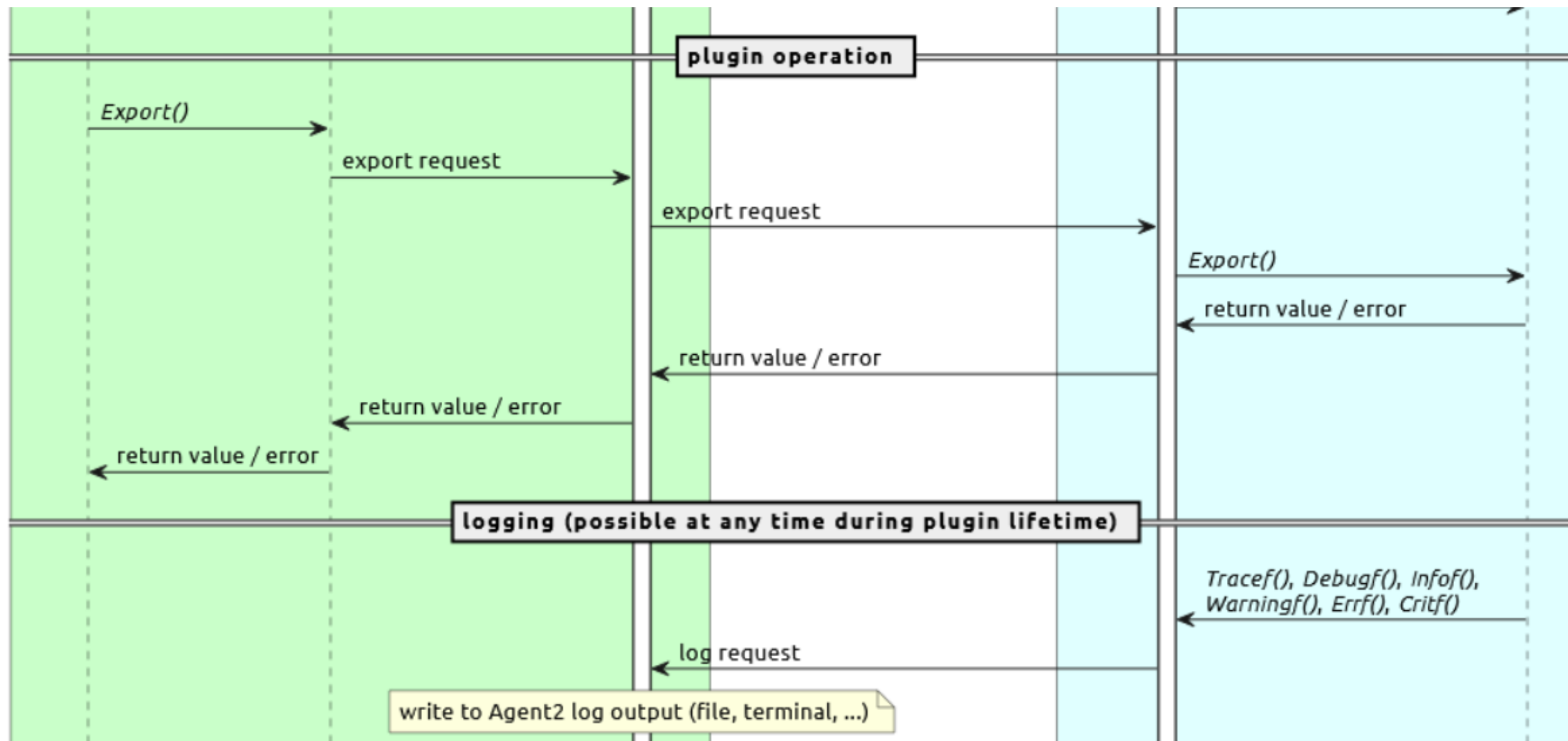
Using Unix sockets or Windows Named pipes.

Zabbix agent 2

Loadable plugin



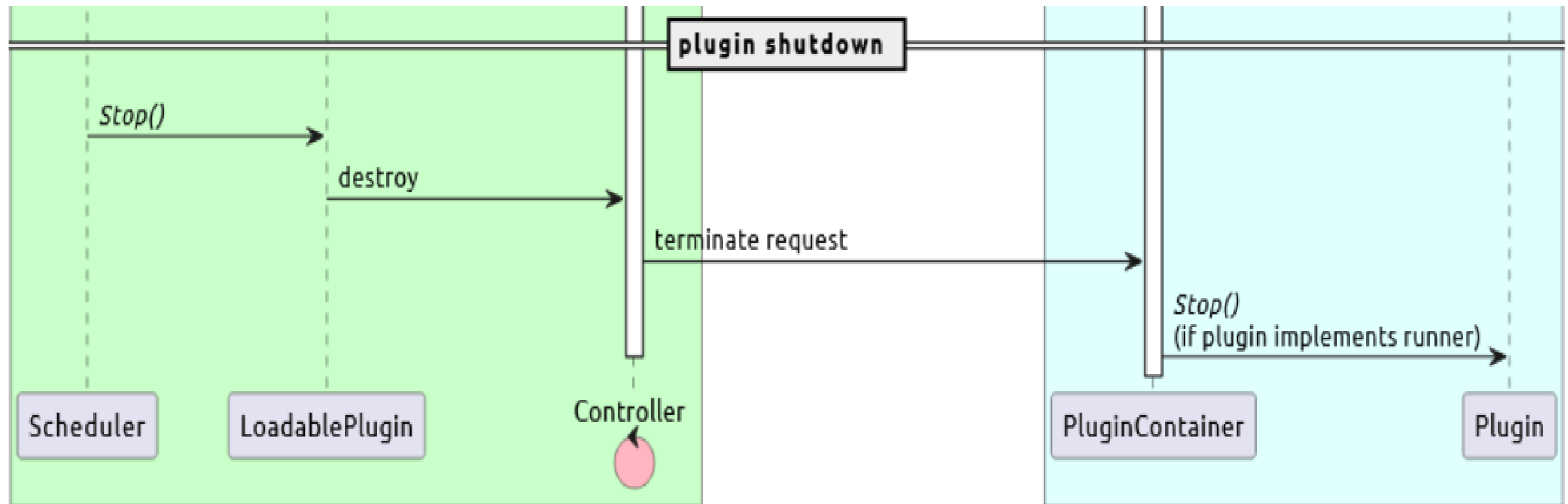
What happens on Zabbix agent 2 start-up?



What happens during plugin lifetime?

Zabbix agent 2

Loadable plugin



What happens on plugin shutdown?

Plugin Interfaces

Configurator

Plugin configuration validation and set-up.

Runner

What to do when plugin starts and stops?

Exporter

This is where all the magic happens, and you can get data.

Zabbix Go SDK

Protocol version

Defines what Zabbix plugin is compatible with what Zabbix version.

Protocol support

Handles all the details of data sending between Zabbix agent 2 and Loadable plugins.

Helper functions

A set of useful tools we use for coding.

In action

Let's run our plugin!

Example plugin

Available on Zabbix git

Has been already cloned! All the useful links at the end of the presentation.

Find it and compile it!

Go to the plugin folder and run "make build."

Configuration

Edit plugin configuration and then Zabbix agent 2 configuration.

Two configuration files to know

```
example.conf x
1  ### Option:Plugins.Example.System.Path
2  #   Path to external plugin executable.
3  #
4  # Mandatory: yes
5  # Default:
6  Plugins.Example.System.Path=/path/to/example/zabbix-agent2-plugin-example
7
8  ### Option: Plugins.Example.Timeout
9  #   Example of a default timeout set in configuration if not , timeout for my ip call
10 #
11 # Mandatory: no
12 # Range: 1-30
13 # Default:
14 # Plugins.Example.Timeout=<Global timeout>
15
16 ### Option: Plugins.Example.Sessions.*.Username
17 #   Example of a session parameter, allowed usernames are hardcoded in code.
18 #   Replace "*" with a session name.
19 #
20 # Mandatory: no
21 # Default:
22 # Plugins.Example.Sessions.*.Username=
23
```

Example plugin configuration

This is where your executable path should be!

```
### Option: TLSCipherAll
#   OpenSSL (TLS 1.2) cipher string.
#   Override the default ciphersuite selection criteria for certificate- and PSK-based encryption.
#   Example:
#       EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
#
# Mandatory: no
# Default:
# TLSCipherAll=

##### Additional configuration files #####

### Option: Include
#   You may include individual files or all files in a directory in the configuration file.
#   Installing Zabbix will create include directory in /usr/local/etc, unless modified during the compile time.
#
# Mandatory: no
# Default:
# Include=

Include=/path/to/example/example.conf

# Include=/usr/local/etc/zabbix_agent2.userparams.conf
# Include=/usr/local/etc/zabbix_agent2.conf.d/
# Include=/usr/local/etc/zabbix_agent2.conf.d/*.conf
```

Zabbix agent 2 configuration

This is where your plugin config path should be!

Run it!

Restart Zabbix agent 2

`"systemctl restart zabbix-agent2"`

Check it?

`"systemctl status zabbix-agent2"` and
`/var/log/zabbix/zabbix_agent2.log`

Get data

`zabbix_get -s 127.0.0.1 -p 10050 -k example.myip`

Custom Loadable plugin

This is what you have been waiting for!

Configure interface in

Example/plugin/config.go

Configure – sets all global plugin configuration options

Validate – checks all provided configuration

```
// Configure implements the Configurator interface.
// Initializes configuration structures.
func (p *examplePlugin) Configure(global *plugin.GlobalOptions, options any) {
    pConfig := &pluginConfig{}

    err := conf.UnmarshalStrict(options, pConfig)
    if err != nil {
        p.Errorf("cannot unmarshal configuration options: %s", err.Error())

        return
    }

    p.config = pConfig

    if p.config.Timeout == 0 {
        p.config.Timeout = global.Timeout
    }
}

// Validate implements the Configurator interface.
// Returns an error if validation of a plugin's configuration is failed.
func (*examplePlugin) Validate(options any) error {
    var opts pluginConfig

    err := conf.UnmarshalStrict(options, &opts)
    if err != nil {
        return errs.Wrap(err, "failed to unmarshal configuration options")
    }

    return nil
}
```

Runner interface in example/plugin/plugin.go

```
// Start starts the example plugin. Is required for plugin to match runner interface.  
func (p *examplePlugin) Start() {  
    p.Infof("Start called")  
}
```

Start

Runs during plugin startup

```
// Stop stops the example plugin. Is required for plugin to match runner interface.  
func (p *examplePlugin) Stop() {  
    p.Infof("Stop called")  
}
```

Stop

Runs during plugin shutdown

Exporter interface in

Example/plugin/plugin.go

```
// Export collects all the metrics.
func (p *examplePlugin) Export(key string, rawParams []string, _ plugin.ContextProvider) (any, error) {
    m, ok := p.metrics[exampleMetricKey(key)]
    if !ok {
        return nil, errs.Wrapf(zbxerr.ErrorUnsupportedMetric, "unknown metric %q", key)
    }

    metricParams, extraParams, hardcodedParams, err := m.metric.EvalParams(rawParams, p.config.Sessions)
    if err != nil {
        return nil, errs.Wrap(err, "failed to evaluate metric parameters")
    }

    err = metric.SetDefaults(metricParams, hardcodedParams, p.config.Default)
    if err != nil {
        return nil, errs.Wrap(err, "failed to set default params")
    }

    ctx, cancel := context.WithTimeout(
        context.Background(),
        time.Duration(p.config.Timeout)*time.Second,
    )
    defer cancel()

    res, err := m.handler(ctx, metricParams, extraParams...)
    if err != nil {
        return nil, errs.Wrap(err, "failed to execute handler")
    }

    return res, nil
}
```

Export – this is where all the magic happens, and data is returned!

Metric registration in

Example/plugin/plugin.go

registerMetrics – all the metrics are set that the plugin will have and that will be sent to Zabbix agent 2 for registration during startup

```
func (p *examplePlugin) registerMetrics() error {
    handler := handlers.New()

    p.metrics = map[exampleMetricKey]*exampleMetric{
        myIPMetric: {
            metric: metric.New(
                "Returns the availability groups.",
                params.Params,
                false,
            ),
            handler: handlers.WithJSONResponse(
                handlers.WithCredentialValidation(
                    handler.MyIP,
                ),
            ),
        },
        goEnvMetric: {
            metric: metric.New(
                "Returns the result rows of a custom query.",
                params.Params,
                true,
            ),
            handler: handlers.WithJSONResponse(
                handlers.WithCredentialValidation(handler.GoEnvironment),
            ),
        },
    }

    metricSet := metric.MetricSet{}

    for k, m := range p.metrics {
        metricSet[string(k)] = m.metric
    }

    err := plugin.RegisterMetrics(p, Name, metricSet.List()...)
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

Handler in

Example/plugin/handlers /handlers.go

```
// MyIP returns the parameters the metric is called as an example.
func (h *Handler) MyIP(ctx context.Context, _ map[string]string, _ ...string) (any, error) {
    req, err := http.NewRequestWithContext(ctx, http.MethodGet, ipifyURL, http.NoBody)
    if err != nil {
        return nil, errs.Wrapf(err, "failed to create request")
    }

    resp, err := h.client.Do(req)
    if err != nil {
        return nil, errs.Wrapf(err, "failed to send the request")
    }

    //nolint:errcheck
    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrapf(err, "failed to read the response")
    }

    return string(body), nil
}
```

MyIP – the actual logic to get your metric, in this case the IP address.

Troubleshooting and more

Let's finish up with a quick overview of how to move forward with your new plugin

To sum it all up!

Key points to take away

Most important takeaways

Architecture

How loadable plugins work! From configuration to execution.

Loadable plugins

Look complicated but can be simplified, we offer support SDK!

Do not be afraid to experiment

Write your own or edit ours! Try a plugin in a different language!

Thank you all!

Time for some questions?

Useful links

Zabbix plugin documentation

<https://www.zabbix.com/documentation/guidelines/en/plugins>

Example plugin repository

<https://git.zabbix.com/projects/AP/repos/example/browse>

Our Plugin Go SDK repository

<https://git.zabbix.com/projects/AP/repos/plugin-support/browse>