

## Runtime Infrastructure - Feature #8544

### Move spawner code to JNI

03/29/2024 02:51 AM - Galya B

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>vendor_id:</b>	GCD
<b>billable:</b>	No		
<b>Description</b>			

#### History

##### #1 - 03/29/2024 03:16 AM - Galya B

The spawner executable exists as a separate deployment with the sole purpose of providing admin rights only to a strictly limited part of FWD functionality. OS user authentication requires native code, so naturally the spawner has been written in C. Its maintenance and development thus requires certain expertise. On execution from the Java server the separate native process poses challenges for the inter-process communication, including argument processing, logging and debugging.

At the same time with distributed infrastructure (deployment with brokers) there is another intermediate process (the broker client) used solely for the purpose of being an interface for the spawner installed on the remote machine. This can be deduced by reviewing the interface BrokerClientServices.

An alternative approach would be to move the spawner code to JNI in a broker client with admin rights.

The pros are:

- validations, logging and overall the spawning process will be driven by Java code;
- less fragile native code (due to the previous point), actual interface with the native code;
- easier maintenance and debugging by more developers.

The cons are:

- This new type of client replacing the broker client and the spawner executable will need to be trimmed down from FWD functionality as much as possible. A new deployment with its own limited classpath will have to be prepared. Ideally the process will only share rpc and authentication mechanics with the core FWD code, so its development may present unexpected challenges.