

# External HTTP Calls

Given that code specified in contracts on SUAVE is executed in TEEs, which can provide attestations about the code they are running, we can more securely make requests to arbitrary http endpoints to fetch data relevant to any offchain computation we are running.

The kinds of data we can fetch are arbitrary : it ranges from price data to ChatGPT responses, or as far afield as your imagination is willing to go.

## Practical Example

In order to understand the implications of fetching data from external sources when running secure, offchain computation in a Kettle, we encourage you to go through our [external calls tutorial](#) . Essentially, we're able to query the ChatGPT completion API with a few lines of code (but this could just as easily be the Binance or Bloomberg APIs too):

```
function
offchain ( )
external
returns
( bytes
memory )
{ bytes
memory keyData = Suave . confidentialRetrieve ( apiKeyRecord , API_KEY ) ; string
memory apiKey =
bytesToString ( keyData ) ; ChatGPT chatgpt =
new
ChatGPT ( apiKey ) ;
ChatGPT . Message [ ]
memory messages =
new
ChatGPT . Message [ ] ( 1 ) ; messages [ 0 ]
= ChatGPT . Message ( ChatGPT . Role . User ,
"Say hello world" ) ;
string
memory data = chatgpt . complete ( messages ) ;
emit
Response ( data ) ;

return abi . encodeWithSelector ( this . onchain . selector ) ; } One additional note: if you wish to create your own Foundry
project with contracts that make external calls, please make sure to adjust thefoundry.toml file to include the following (the
equivalent of runningsuave-geth with the--suave.eth.external-whitelist='*' flag):

[ profile.suave ] whitelist =

[ "*" ]
```

## Interface

SUAVE exposes one precompile which can handle all your external call needs:

function

doHTTPRequest ( HttpRequest memory request )

internal

returns

( bytes

memory ) You can see how this is used in everything from [EthJsonRPC](#) (which powers patterns like [Gateway.sol](#) ), to [sending bundles to the Flashbots relay](#) , to querying [ChatGPT.sol](#) . [Edit this page](#) [Previous Block Building](#) [Next MEV Supply Chain Interface](#)