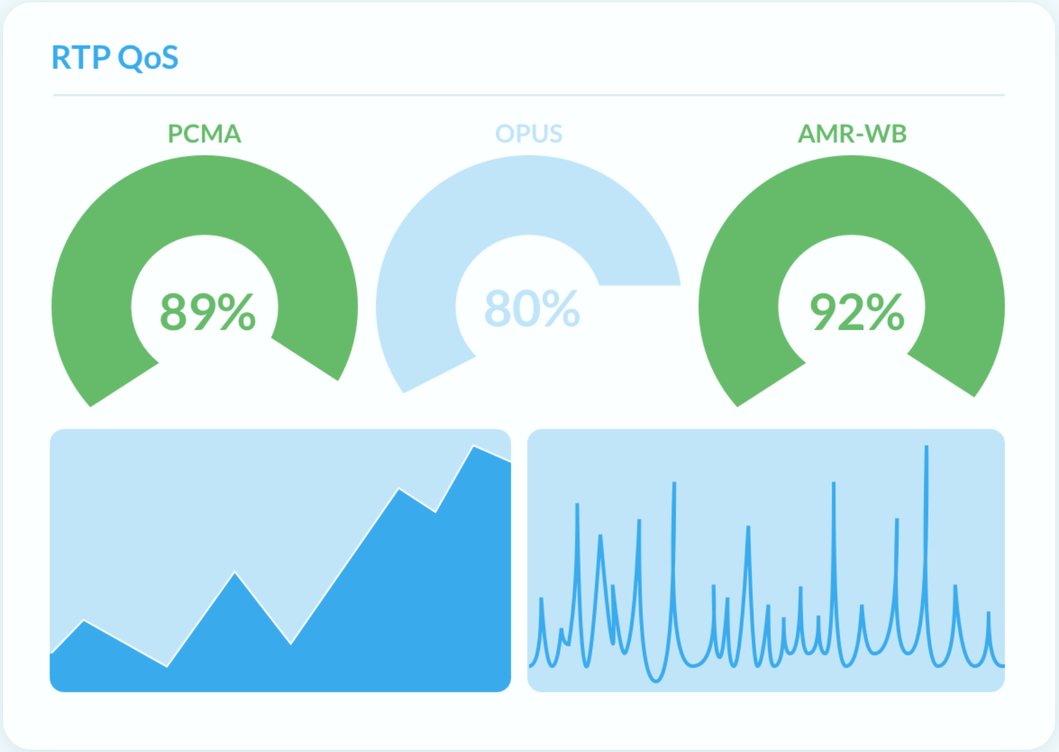
How to build SIP3 based solutions or Wangiri fraud detection example

Oleg Agafonov
FOSDEM’2021

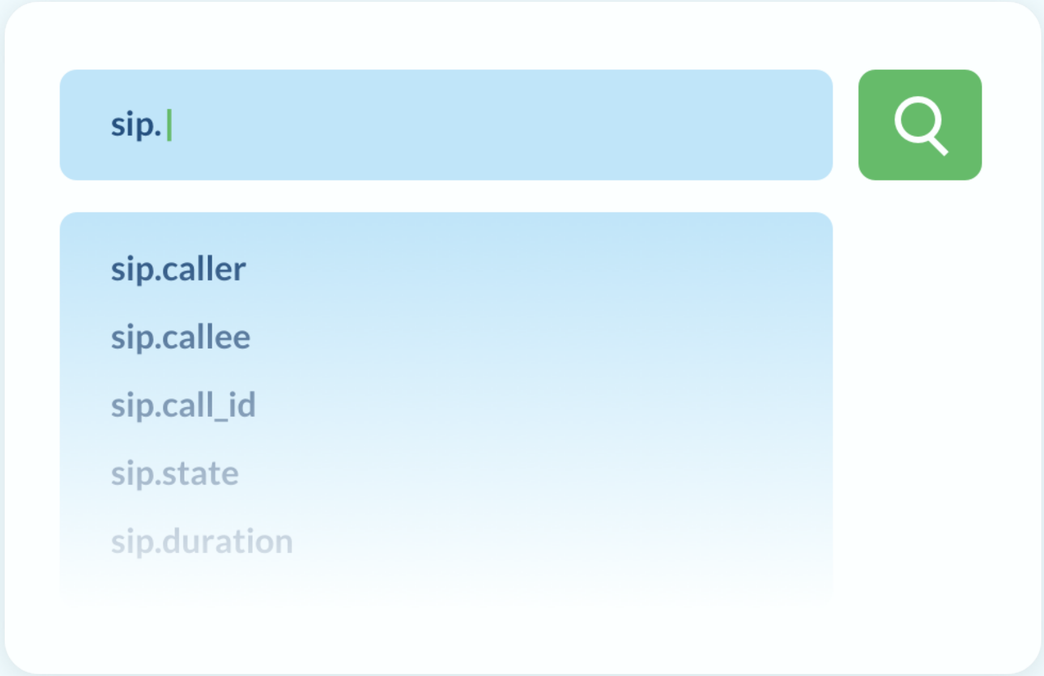
Monitoring

- SIP and RTCP/RTP QoS metrics
- Multi-dimensional metrics
- Integrations with multiple monitoring platforms



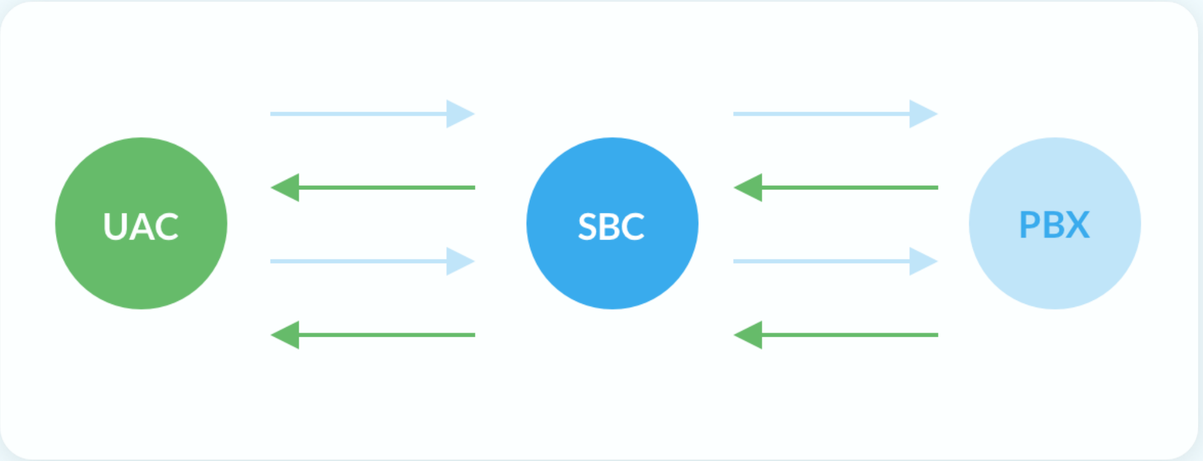
Search

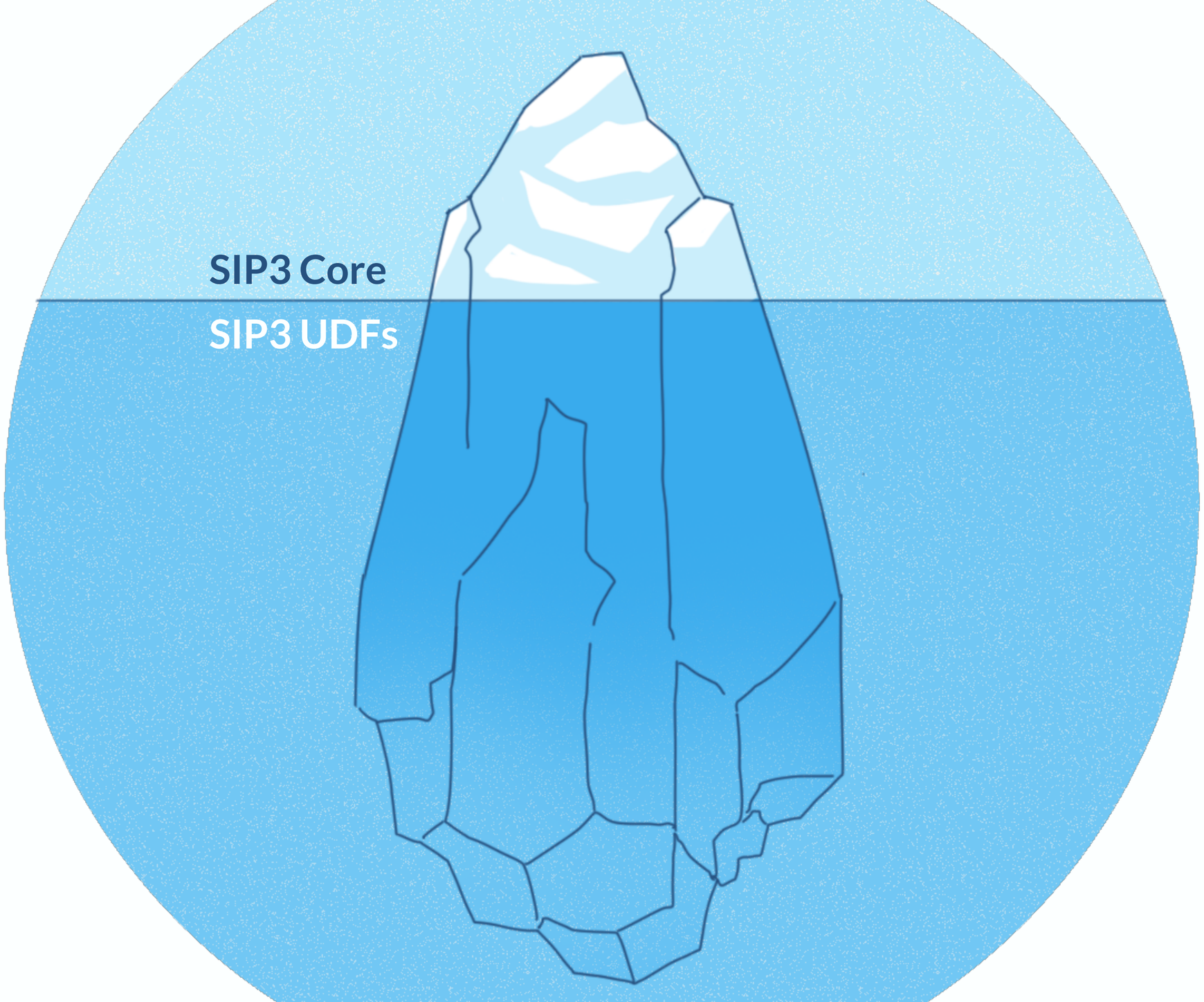
- ip, sip, rtcp and rtp groupings of search attributes
- =, <, >, !=, ~= search operators
- Complex search queries



Call Details

- United SIP and RTCP/RTP call flow diagrams
- Advanced call correlation logic
- Granulated media QoS reports
- Export to .pcap



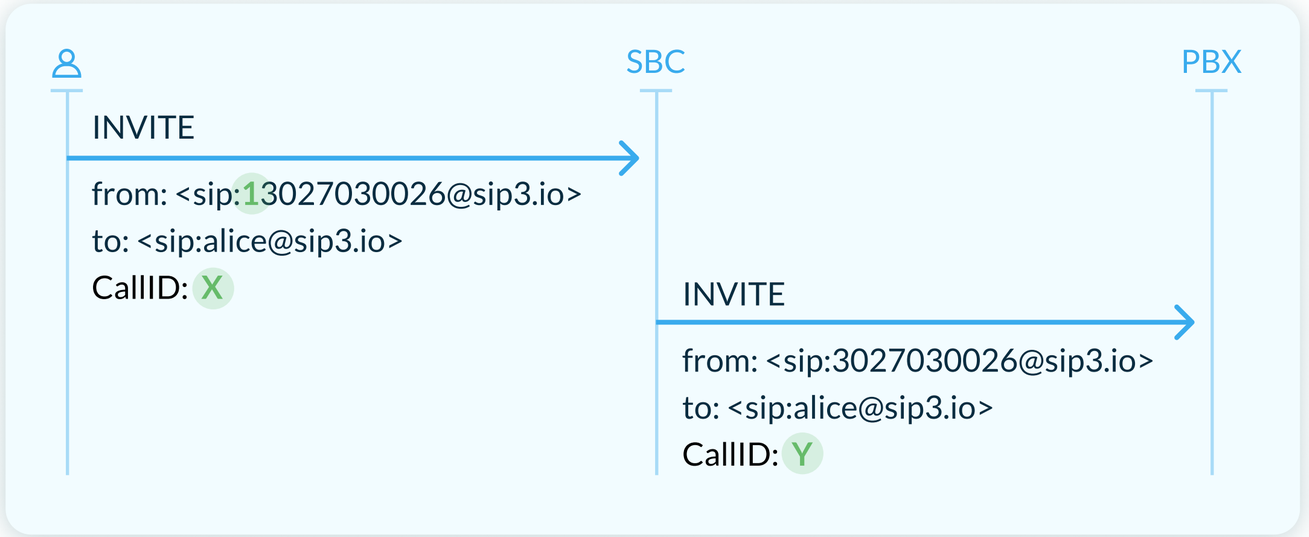

SIP3 Core
SIP3 UDFs

part I

sip_message_udf

Call correlation


```
package udf

def eventBus = vertx.eventBus()
eventBus.localConsumer("sip_message_udf", { event ->
    def packet = event.body()

    def from_header = packet['payload']['from']
    def matcher = (from_header =~ /1(\d*)/)
    if (matcher) {
        packet['attributes']['caller'] = matcher[0][1]
    }

    event.reply(true)
})
```



Adding `robocall` attribute


```
package udf

def eventBus = vertx.eventBus()
eventBus.localConsumer("sip_message_udf", { event ->
    def packet = event.body()

    def sip_message = packet['payload']
    if (sip_message['from'].matches('<sip:100@.*')) {
        packet['attributes']['robocall'] = true
    }

    event.reply(true)
})
```

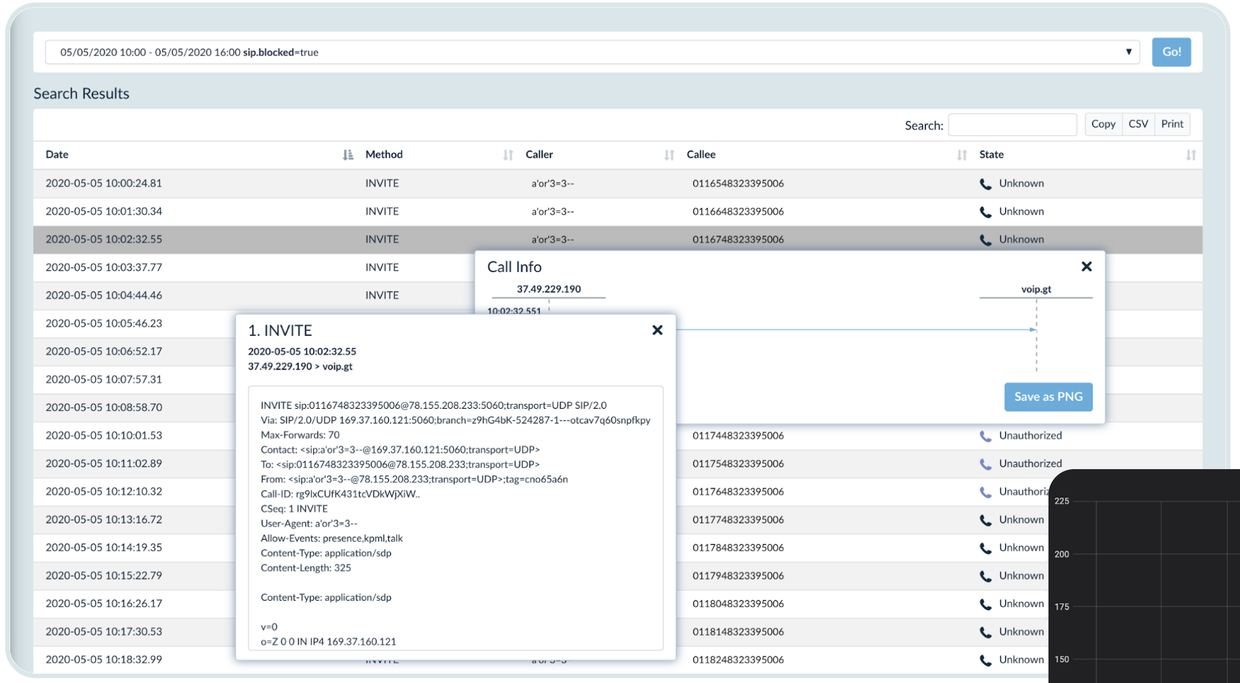
Parsing STIR/SHAKEN

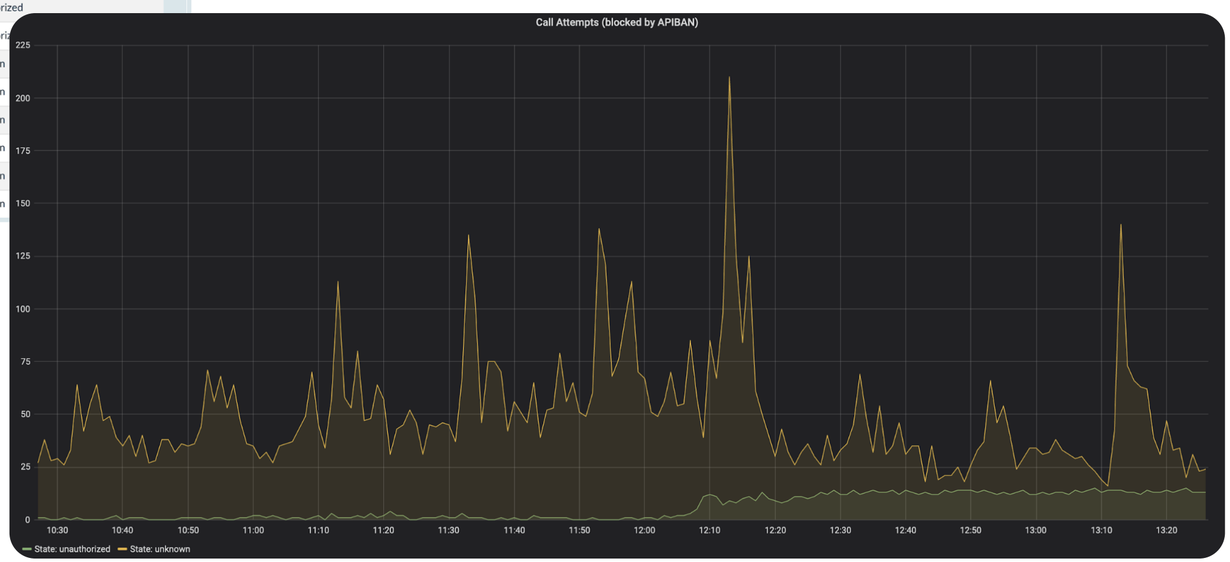

```
package udf

import io.vertx.core.json.JsonObject

def BASE64_DECODER = Base64.decoder

def eventBus = vertx.eventBus()
eventBus.localConsumer("sip_message_udf", { event ->
    def packet = event.body()

    def sip_message = packet['payload']
    def request_line = sip_message['request-line']

    if (request_line != null && request_line.startsWith('INVITE')) {
        def identity_header = sip_message['identity']
        if (identity_header != null) {
            def payload = identity_header
                    .split(';')[0]
                    .split('\\.')[1]
                    .getBytes()

            def identity = new JsonObject(new String(BASE64_DECODER.decode(payload)))
            packet['attributes']['attest'] = identity.getString('attest')
        }
    }
    event.reply(true)
})
```

Integrating with APIBAN


```
package udf

import io.vertx.core.AbstractVerticle

class ApiBanUdfExample extends AbstractVerticle {

    private Set blocked = []

    @Override
    void start() throws Exception {
        def eventBus = vertx.eventBus()
        eventBus.localConsumer("sip_message_udf") { event ->
            def packet = event.body()

            if (blocked.contains(packet['src_addr'])) {
                packet['attributes']['blocked'] = true
                packet['attributes']['blocked_addr'] = packet['src_addr']
            }

            event.reply(true)
        }

        asyncUpdateBlocked()
    }
}
```

Integrating with APIBAN





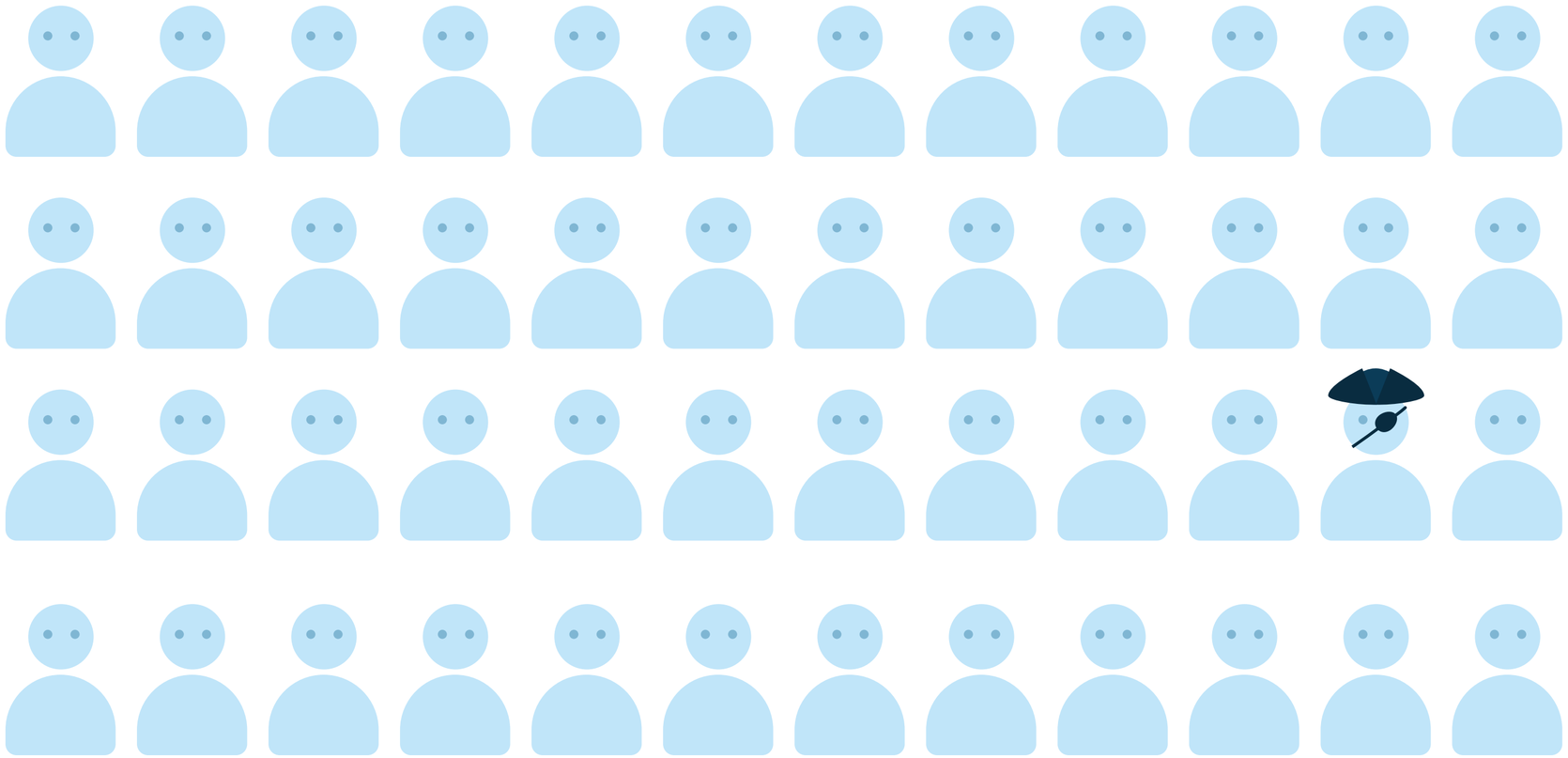
part II

sip_call_udf

Adding `problematic` attribute


```
package udf

def eventBus = vertx.eventBus()
eventBus.localConsumer("sip_call_udf", { event ->
    def session = event.body()

    def setup_time = session['payload']['setup_time']
    if (setup_time != null && setup_time > 5000) {
        session['attributes']['problematic'] = true
    }

    event.reply(true)
})
```

Use Case



SIP3 solution for Wangiri fraud detection

sip_call_udf


```
package udf

import io.vertx.core.AbstractVerticle
import io.vertx.core.json.Json

class SipCallUdfHandler extends AbstractVerticle {
    def udp = vertx.createDatagramSocket()

    @Override
    void start() throws Exception {
        def eventBus = vertx.eventBus()
        eventBus.localConsumer("sip_call_udf") { event ->
            // Encode a CDR as JSON
            def buffer = Json.encodeToBuffer(event.body())

            // Send the encoded CDR via UDP to 127.0.0.1:15080
            udp.send(buffer, 15080, "127.0.0.1") {}

            event.reply(true)
        }
    }
}
```

Profiler


```
class Profile {

    val outgoingCallStats = CallStats()
    val incomingCallStats = CallStats()

    class CallStats {

        var totalCalls = 0
        var totalDuration = 0
        var chargedMinutes = 0

        var failedCalls = 0
        var canceledCalls = 0
        var answeredCalls = 0

        var terminatedCalls = 0
        var threeSecondsCalls = 0
    }
}
```

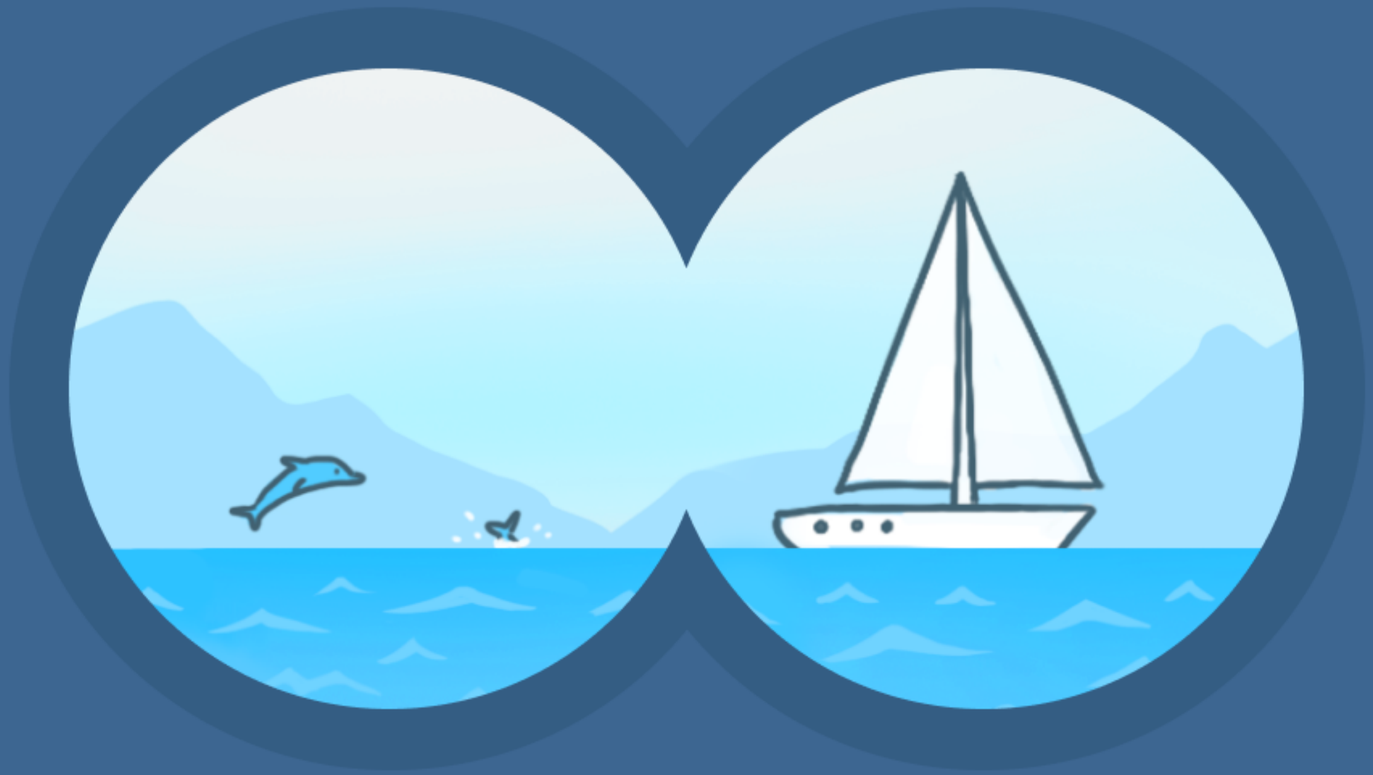
part II

sip_call_udf

msisdn	totalCalls	chargedMinutes	failedCalls	canceled(	answered(	terminate(	threeSecon	terminatedCa	threeSeconds(
*******6323	241263	1021	233724	1247	967	558	119	0.5770423992	0.1230610134
*******5744	240934	1170	233509	971	1042	588	104	0.5642994242	0.09980806142
*******7512	239514	999	231867	1460	942	539	119	0.5721868365	0.1263269639
*******5879	239184	1447	231783	1100	1162	623	103	0.5361445783	0.08864027539
*******8449	238093	1209	230424	1283	1034	585	126	0.5657640232	0.1218568665
*******0273	236951	1005	229352	1591	933	531	120	0.5691318328	0.1286173633
*******6983	236549	935	228934	1650	894	498	104	0.5570469799	0.1163310962
*******1056	236539	1582	229355	857	1295	706	134	0.5451737452	0.1034749035
*******1027	235102	1222	227536	1386	1022	572	121	0.5596868885	0.1183953033
*******6069	234078	1360	226859	964	1080	596	95	0.5518518519	0.08796296296
*******4405	146149	1051	137355	2136	992	530	112	0.5342741935	0.1129032258
*******8265	116673	10528	75642	25415	11743	8499	2225	0.7237503193	0.1894745806
*******8518	77717	1240	73684	368	1281	65	36	0.05074160812	0.0281030445
*******4971	77657	1277	73663	364	1325	57	47	0.04301886792	0.03547169811
*******7688	77569	1206	73700	374	1239	67	34	0.05407586764	0.02744148507
*******8583	77248	1285	73335	366	1314	70	29	0.05327245053	0.02207001522
*******6604	76853	1310	72917	332	1342	61	37	0.04545454545	0.02757078987
*******5160	76506	1261	72420	405	1315	69	45	0.05247148289	0.03422053232
*******5141	76096	1269	71968	362	1306	59	43	0.04517611026	0.03292496172
*******3961	73844	1223	70038	365	1270	85	38	0.06692913386	0.02992125984
*******9708	73776	5	30106	34884	4739	4718	4680	0.9955686854	0.9875501161
*******4811	67452	0	63051	1345	783	780	775	0.9961685824	0.9897828863
*******0684	63135	0	60172	952	1	1	1	1	1

msisdn	totalCalls	chargedMinutes	failedCalls	canceled(	answered(	terminate	threeSecon	terminatedCa	threeSeconds(
*******4392	169	0	23	132	5	5	5	1	1
*******8489	167	0	9	142	6	6	6	1	1
*******9100	27	0	6	2	13	13	13	1	1
*******9296	780	0	60	653	30	30	30	1	1
*******4803	70	0	0	36	25	25	25	1	1
*******1797	5607	0	2810	2485	29	29	29	1	1
*******3117	73	0	17	0	39	39	39	1	1
*******1739	5595	0	2783	2485	35	35	35	1	1
*******1760	5607	0	2814	2455	52	52	52	1	1
*******4738	3707	0	694	2712	107	107	107	1	1
*******7550	5614	0	2801	2466	43	43	43	1	1
*******1612	255	0	78	115	44	44	44	1	1
*******9205	771	0	83	633	30	30	30	1	1
*******2342	494	0	60	403	17	17	17	1	1
*******1808	5618	0	2807	2454	54	54	54	1	1
*******9714	5603	0	2808	2453	37	37	37	1	1
*******5073	239	0	88	101	40	40	40	1	1

Thank you!



Visit us at sip3.io
or github.com/sip3io

And join our community channels in Slack and Telegram