

### Introduction to SpinBoard

What is SpinBoard?



• **SpinBoard** is a highly customizable, Web based, low-code, aggregation and visualization platform for heterogeneous data.

- *SpinBoard* is built with two main concepts in mind:
  - Provide the user with an extremely intuitive, self-configurable interface
  - Easily allow the integration of huge amounts of data, coming from heterogeneous sources

### Introduction to SpinBoard

Widgets as the main concept for the users



- SpinBoard provides a set of widgets to display data
- Every user can arrange the widgets as preferred:
  - the system automatically preserves the interface status and layout
  - two users can display the same data in a completely different way, according to their preference.
- SpinBoard is designed to integrate and display streams of data and provides a set of ready to use data integration flows
  - when the integration of a new stream is needed, little or no java code is needed

## SpinBoard Goals

Born to meet the customer needs



**SpinBoard** was born to meet the specific customer need to monitor and manage a complex system with as low as possible effort:

- Providing a real-time system status overview
- Providing support information for investigation and analysis
- Enabling pro-active issues management
- Integrating existing instruments and software

All the instruments already in place are providing needed information, but in a fragmented way, thus resulting in high investigations times and a lot of skills required for the job.

### **Functionalities Comparison**

SpinBoard compared to other monitoring tools



#### Following a brief list of the most used / necessary functionalities

Functionality	Geneos	Splunk	SpinBoard	Notes
gRPC Connector	Ν	Y	Y	
ZMQ Connector	Ν	Y	Y	
Aeron Connector	Ν	Ν	Y	
Custom Localization	Ν	N	Y	SpinBoard supports custom translations
Custom Data Segregation	Ν	Ν	Y	
Hierarchical Grids	Ν	Ν	Y	
Grids filtering / sorting	Ν	Y	Y	Splunk requires query modification
Cloud Ready	Ν	Y	Y	
Task Execution	Y	N	Y	
Oracle/sql	Ν	Y	Y	Splunk lacks custom data mapping
Real time data streaming	N	N	Y	

# Other possible tools Tools similar to SpinBoard.



Functionality	Qlik Sense	Tableau	PowerBl	SpinBoard	Notes
gRPC Connector	N	N	Y	Y	• PowerBI requires to buy a
Kafka Connector	N	N	Y	Y	third party connector
Aeron Connector	N	N	Ν	Y	• PowerBI data can
ZMQ Connector	N	Ν	Ν	Y	be pushed via rest api
Real time data streaming	Y	N	N	Y	subscribing
Oracle/sql Connector	Y	Y	Y	Y	
Custom Data Segregation	N	Y	Y	Y	
Hierarchical Grids	Y	Y	Y	Y	
Grids filtering / sorting	Y	Y	Y	Y	
Cloud Ready	Y	Y	Y	Y	

#### A layered tool One tool, different audience





11

#### **SpinBoard Use Cases** SpinBoard use case #1 : daily orders monitoring



Monitoring Orders status, intercept errors and investigate the cause:

- Before *SpinBoard*, PS Team had to:
  - Configure an alert on **Splunk**
  - Open **SQL Developer**, write the very specific query for extracting order data
  - On Splunk, rewrite the query for investigating the specific cause
- With *SpinBoard*, PS Team can now:
  - Immediately see in blotters any problematic order. All data is already presented there, click on a cell to open a dialog containing a Splunk query ready to be checked

### What does integration really mean?



≡	PS-Monitoring	T080 ×	TOBO - Brand	hes X [	Desktop 3 ×	Desi	ktop 4 × Desktop 5 >	× Desktop 6 ×	Desktop 7 ×	Desktop 8	× +									*
Trader blotter							<b>\$</b> ×	Feed Jobs Status												<b>\$</b> ×
Date	† <sub>1</sub> Internalid †	с ţ	Status †	Legal E… ↑	Cur †₊	T. †	Splunk Link † E:	dof	† <sub>4</sub> S	†₄ Thread			t₁ Time		ţ	Next Run		†1	Error Message	†4
2022-08-01 12:20:3	8 1891503210/ŀ SP_L	TP	CONFIRMED	-MUNICH	EURUSD	SPOT	1891503210/HOSP_L_/	C02-FEED	ENI	JOBS_EX	ECUTOR7		2022-08	-02 09:45:00,303	3	2022-08-021	10:45			
2022-08-01 12:20:54	4 1891503310/F SP_L	TP	CONFIRMED	-MUNICH	EURUSD	SPOT	1891503310/HOSP_L_/ A	CC-CONFIG-CHECKER	ENI	JOBS_EX	ECUTORO		2022-08	-02 07:12:18,63	5	2022-08-031	03:20			
Comple				Vafl	-	+h	1891503410/HOSP_L_/	CORPORATE-PORTAL-SPA	ENI	JOBS_EX	ECUTOR4		2022-08	-02 06:04:14,787	7	2022-08-031	06:00			
Comple	ex Splunk	que	ery on	Nalk	a wi	un	1891503510/LMAX_L_# A	CZR-FEED	ENI	JOBS_EX	ECUTOR6		C	omnl		nlun	k au o	~	visualico	daca
custom	protobu	fpro	ocessi	ng			1891503610/LMAX_L_A A	EVENTS-SYNCHRONIZER	ENI	JOBS_EX	ECUTOR5			ompi	ex S	pium	k que	۱y	visualise	u as a
Custom	procosa	· p·					1891503710/HOSP_L_/ A	INBOX-CLEANUP	ENI	JOBS_EX	ECUTORO		ta ta	able v	vith	cust	om co	blo	r status	
2022-08-01 12:44:22	2 1891503810/F SP_L	TP	CONFIRMED	-MUNICH	EURUSD	SPOT	1891503810/HOSP_L_/ A	LEI-WARNING-EMAIL	ENI	JOBS_EX	ECUTOR2									
2022-08-01 13:00:1	.3 1891503910/F SP_L	TP	CONFIRMED	-MUNICH	EURUSD	SPOT	1891503910/HOSP_L_/ A	MARIO	ERF	OR JOBS_EX	CUTORO		2022-08	-02 02:02:09,069	9	2022-08-031	02:00		eu.unicredit.exceed.cockpi	t.mario.exceptic
2022-08-01 13:01:0	0 1801504010/IMAX I Splunk Link: 1891503210	TD /HOSP_L_AH/	/TRD_PROXY/EUR/	_MUNICH	FLIDGRD	SDOT	1801504010/LMAY L / A	MTD_TDANED_NATA	FDD				2022-09	-02 03:02:07,366	6	2022-08-031	03:00		org.springframework.web.	client.Resource#
2022-08-01 13:01:0	( Id 1, Timest 1,	Text										5 t <sub>1</sub>	Environment	02 10:13:18,656	6	2022-08-021	12:05			
2022-08-01 13:01:0	1	[INFO ] 20	22-08-01 10:20:38	8.103 :: (api-akka	actor.default-	dispatcher-	-21] :: e.a.s.blotter.TradesKafkaC	onsumer @ /user/trades-con:	sumer Sending a n	otification for tr	de			02 09:28:24,651	1	2022-08-021	11:10			
2022-08-01 13:01:07	2	18915032	210/1 /TF	RD_PROXY/EUR/	USD/85XMFZ	REHA to O si	ubscribers. trade: external_id: "/	AH5W8NKE00031"~~~~intern	al_id:					)2 08:21:34,869	5	2022-08-021	10:20			
2022-08-01 13:01:07	2	trading_re	ceived_at {~~~~ t	RD_PROXY/EUR	/USD/85XMFZ	REHA ~~~[	iming {~~~ api_receiveo_at {~~	~ timezone: "o it."~~~ times	tamp: 165934923	JS:				02 09:50:15,531	1	2022-08-021	11:50			
Booking & Trading So	2022-08-01	CONFIRME {~~~ vear:	ED~~~deal_type: 2022~~~ mont	Para	met	ric S	Splunk qu	ery		POT"~~~~ 186110	value_date	~ api-1	uat							<b>\$</b> ×
Query	12:20:39	downstrea	am_id: "Not bookee			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	egs ( account_10. 501	tegor_entity.		enor: "SP	OT"~~~									
1 SELECT * I	F	value_date 11861100	e {~~~ year: 2022~ )000~~~ }~~~ dow	~~ month: 8~~ nstream_id: "No	~ day: 3~~~ }^ )t booked''~~~	~~~ base {~ booking_st	~~~ value: 500000.0~~~~ side: 56 tatus: BOOKED~~~~ }~~~~ details	<pre>{ term { value { base_currency: "EUR"</pre>	~~ term_currency:	"USD"~~~ spot,	.: precision: 5~~	~		L						0
ID	1	}~~~~}~~~~( "TP"~~~~}~	counterparty {~~~~ a	app_id: "AUTOHE	EDGER"~~~ cli tratedv: "Take	ent: "	"~~~~ trader_user_name: "	'899"~~~ desk_id: "AG-MUNI	ICH-AUTOHEDGE"~	channel_id:				1 1 <u>1</u>	APP_ID	t⊥	CLIENT	t⊥	CLIENT_INTERNAL †	CLIENT_FULI
219271		TINEO 1 20	22.09.01.10.20.29		actor default	dicpatchor	161 o o c blottorTrodocKoR/oC	acumar @ lucar/tradar con	sumer Conding a p	otification for tr	de				UnicreditUBI	ISIntegratio	Russia Sales Des	k	null	Russia Sales
219280	131/4/0100313703		22-08-01 10.20.30 30000042		LactoLOPIAUL	iou iou	nou	nou nou	onoenn	eu	001		CULIVI		UnicreditUBI	ISIntegratio	LNG10		null	LNG10
219251	1513681513635.AG.V	/in LEL74B	0000025	CONFIRMED	r	าบไไ	null	null	undefin	ed	UCT		CLIENT		UnicreditUBI	ISIntegratio	LNG10		null	LNG10
220153	1526367107587.AG.	/in AQ30B	C0000136	REJECTED		<b>^</b>							CLIENT		UnicreditUBI	ISIntegratio	LNG10		null	null
229683	1548310134000.650	67 606TO	C0001909	CONFIRMED		Con	nplex cust	tom SQL	query				CLIENT		UnicreditUBI	ISIntegratio	69777		null	650670
220154	1526541601213.AG.	/in AQ30BI	C0000137	CONFIRMED		IUII	nott	nou	onoenn	20	001	_	CLIENT		UnicreditUBI	ISIntegratio	LNG10		null	LNG10
220155	1526541601214.AG.V	/in AQ30BI	C0000138	CONFIRMED	r	าบไไ	null	null	undefin	ed	UCT		CLIENT		UnicreditUBI	ISIntegratio	LNG10		null	LNG10
-220156	1526541601215 AGV	An AOBOR	C0000139	CONFIRMED	r	null	null	null	undefin	ed	UCT		CLIENT		UnicreditUBI	ISIntegratio	ING10		null	ING10 -

#### What does integration really mean? Integration benefits



- Centralize different applications in a single point:
  - Instead of opening Splunk, Geneos, SQL Developer .. all needed functionalities are grouped together in a single FE
- Transparently integrate heterogeneous information sources:
  - No matter from where information comes from: it is always represented in the same way
  - All grids have the same controls, filtering, sorting and grouping always work in the same way

#### Integration benefits Why Integrating



#### • Simplified decision making

Working on a unified view simplifies the decision making process, removing the need to move between different applications to access data

#### • Enhanced analysis

Related data is more meaningful & powerful when it is pulled together in one application. Analysis of multiple data sources is better handled by bringing the data together where trends and conclusions can be drawn much sooner

SpinBoard use case #2 : daily checks / post release checks

Pin

Monitoring platform status and each service functionalities :

- Before *SpinBoard*, PS Team had to:
  - Check in Geneos each service status
  - Check on **Splunk** specific dashboards
  - Open **UCT** login page and manually verify
  - Open, one by one, services REST endpoints to analyse returned data
- With *SpinBoard*, PS Team can now:
  - Just open the "System Status" tools and check

#### SpinBoard use case #2 : daily checks / post release checks



	PS-Monitoring sy	stem status 🕂											•	
⊞	Alert Log blotter	ExCEED Status											<b>\$</b> ×	
		App Name						Health Check					t↓	
•	Booking & Trading Sql Widget	✓ Api-*						OK / OK / OK						
田	Client blotter	api-1						OK						
		api-2						ОК						
•	Cockpit Sql Widget	api-3						ОК						
œ	Commodity Blotter	Booking						ОК						
	commonly blotter	Credit-Check						ОК						
•	Config db Sql Widget	D3-Proxy						ОК					l	
m	Core pricing Blotter	Pricing						ОК						
	core pricing blotter	✓ Pricing-Custom-*						OK / OK						
▦	Status							017	_			_	-	
	Cool John Chatra	Feed Jobs Status	•	Chaburg	•	Thursd	•		Next Dec	•	-		¢ ×	
	Feed Jobs Status	00	'ţ	Status	't		't		Next Run	't	Error N	Message	't	
▦	Risk blotter	·FEED		SIARI		JOBS_EXECUTOR4		2022-08-27 01:10:00,000	2022-08-27103:10					
		EVEN IS-SYNCHRONIZER		END		JOBS_EXECUTOR9		2022-08-27 01:01:19,214	2022-08-27101:35					
>	Splunk widget	LO2-FEED		END		JOBS_EXECUTOR8		2022-08-27 00:46:16,129	2022-08-27101:45					
囲	Trader Logout	-NY-FEED		END		JOBS_EXECUTOR3		2022-08-27 00:40:02,993	2022-08-27102:40					
	5	CZR-FEED		END				2022-08-27 00:28:55,241	2022-08-27101:20					
	Trader blotter	-NY-FEED		END		JOBS_EXECUTORS		2022-08-27 00:21:43,817	2022-08-27102:20			and a star second second		
		U7-FEED		EKKOR		JOR2_EXECUIOK8		2022-08-27 00:05:40,002	2022-08-27102:05		org.sp	oringframework.tra	ansaction.C	
		-FEED		ERROR		JOBS_EXECUTOR2		2022-08-26 23:50:30,059	2022-08-27T01:50		org.sp	pringframework.tra	ansaction.C	

SpinBoard use case #3 : commodity deals blotter



Monitoring the deals on commodities:

- Before *SpinBoard*, Sales:
  - Does not have any possibility to see these data on any tool.
  - The only solution is to implement and integrate a new blotter using other vendor-provided front-ends

#### • With *SpinBoard*, Sales:

- Can just look at the commodity blotter containing all the information related to commodity deals.
- This blotter filter out the information based on the Legal entity of the sale (which is necessary for regulatory needs)

#### SpinBoard use case #3 : commodity deals blotter



=	PS-Monitoring Syst	tem status × Com	modity blotter 🛛 🗙	+					<b>.</b>				
Ħ	Alert Log blotter	Commodity Blotter											
		Exceed time $~~ \downarrow~$	Platform 🗘	Symbol †	Deal type 👌	Currency †	Commodity Id 🗘	External Id 1	Stream type				
•	Booking & Trading Sql Widget	2022-08-29 14:51:51	UCT	LEAD CASH	ASIAN_SWAP	USD	3016	COMM3016	RFS				
⊞	Client blotter	2022-08-29 14:51:06	UCT	LEAD CASH	ASIAN_SWAP	USD	3016	COMM3016	RFS				
ш		2022-08-29 14:51:03	UCT	LEAD CASH	ASIAN_SWAP	USD	3016	COMM3016	RFS				
•	Cockpit Sql Widget	2022-08-29 14:51:03	UCT	LEAD CASH	ASIAN_SWAP	USD	3016	COMM3016	RFS				
m	Commodity Plotter	2022-08-29 14:51:02	UCT	LEAD CASH	ASIAN_SWAP	USD	3016	COMM3016	RFS				
ш		2022-08-29 14:50:45	UCT	LEAD CASH	ASIAN_SWAP	USD	3016	COMM3016	RFS				
•	Config db Sql Widget	2022-08-29 14:50:43	UCT	LEAD CASH	ASIAN_SWAP	USD	3016	COMM3016	RFS				
m		2022-08-29 14:22:37	UCT	NICKEL CASH	ASIAN_SWAP	EUR	3015	COMM3015	RFS				
<b></b>	Core pricing Blotter	2022-08-29 14:22:35	UCT	NICKEL CASH	ASIAN_SWAP	EUR	3015	COMM3015	RFS				
⊞	Status	2022-08-26 15:49:36	UCT	EUA FINANCIAL	BULLET	EUR	3014	COMM3014	RFS				
_		2022-08-26 15:49:36	UCT	EUA FINANCIAL	BULLET	EUR	3014	COMM3014	RFS				
⊞	Feed Jobs Status	2022-08-26 15:48:46	UCT	NYMEX NG HENRY HUB	BULLET	USD	3013	COMM3013	RFS				
⊞	Risk blotter	2022-08-26 15:48:46	UCT	NYMEX NG HENRY HUB	BULLET	USD	3013	COMM3013	RFS				

#### Splunk widget

SpinBoard use case #4 : daily business trades monitoring



Monitoring platform hedging trades:

- Before *SpinBoard*, traders had:
  - A flat UCT blotter, hard to inspect and missing some information that can be integrated using other vendor-provided front-ends
- With *SpinBoard*, Traders will be able to:
  - Just look at a multi-source blotter aggregating trades from kafka, trade-capture deals stored in the DB and tradepoint-client messages from a different kafka topic. All data represented in a single multi-hierarchical blotter

### SpinBoard Architecture

A general overview of the SpinBoard architecture





#### SpinBoard Architecture Backend structure



Backend can be split into five different sub-components:

- **1. Source** fetches the data from an external component
- 2. Adapter converts the data to an internal representation
- 3. Core stores and merges the data from different sources
- 4. DataFilter filters data for the user view
- 5. Target sends the data to the front end

Sub-components just need to be initialized, clients can write their own specific sub-components extending proper interfaces. FE widgets are the result of a chain or a composition of chains built using above sub-components.

### SpinBoard Architecture

The chain structure of the backend



A general setup containing two different sources and three different widget can be modelled using the following graph. External information Adapter 1 Core 1 DataFilter 1 Widget 1 Source 1 Adapter 2 Core 2 Widget 2 </> Source 2 SpinBoard UI Adapter 3 widget 3 Core 3

- Widget 1 gets the data from Source 1 through Adapter 1 with a data filter layer.
- Widget 2 contains data from Sources 1 and 2 through respectively Adapters 1 and 2
- Widget 3 gets the data from Source 2

### SpinBoard Architecture

The adapter as a stateless mapping component



The goal of an adapter is to map the data from an external source to an internal representation.

The tool provides a general Interface receiving a generic EventMessage containing the data from a source and returning a list of Rows.

```
public interface Adapter {
    List<Row> baseAdapt(EventMessage<?> message);
```

Every customer can implement a set of adapters based on the widget requirements:

```
private List<Row> buildRow(EventMessage<KafkaMessage> message)
    Map<Long, Cell> cells = new HashMap<Long, Cell>();
```

For every KafkaMessage a row containing three cells is created

cells.put(NAME, CellBuilder.builder().withDisplayValue(message.getName()).build()); cells.put(ID, CellBuilder.builder().withDisplayValue(message.getId()).build()); cells.put(PRICE, CellBuilder.builder().withDisplayValue(message.getPrice()).build());

Row finalRow = newRowBuilder().withId(message.getId()).withAction(INSERT).withCells(cells).build();

```
return Lists.newArrayList(finalRow);
```

Then the resulting row is returned to the tool for the next step of the process

#### Thank You!

Contact us at:

Info@t-spin.com

Visit us at:

https://www.t-spin.com