Integration Analytics
(You can’t fix what you can’t see.)

Russ Teubner, Distinguished Engineer – Application Modernization

(Co-Founder and CEO of HostBridge Technology)

Greg Smith, Software Engineer

(Splunk SME, Data Analytics)

Last Updated: Sept 13, 2022

BROADCOM SOFTWARE

Agenda

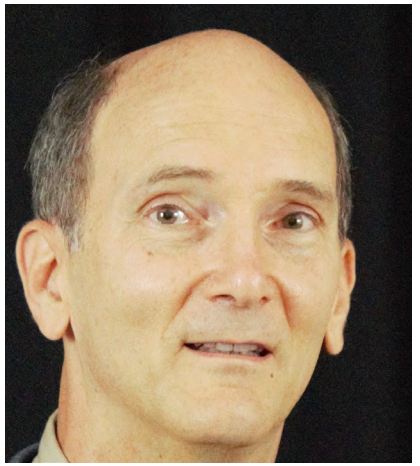
- Speaker Bio
- Integration Analytics… Why?!
- Overview of HTAC and Macro Hunter
- HTAC and Macro Hunter: Behind the Scenes
- HTAC Macro Hunter Dashboards
- Live Demo
- Bad Bot Analysis
- Q/A

BROADCOM SOFTWARE

Speaker Bios

Russ Teubner

Russ is a Distinguished Engineer focusing on mainframe application modernization. A seasoned inventor and entrepreneur, over the last 40 years Russ has applied his creative energies to solving difficult problems associated with integrating IBM mainframes and emerging technologies. As the CEO and co-founder of HostBridge Technology, Russ positioned its flagship integration platform, HB.js, as a solution for large global enterprises to bridge the gap between hybrid/cloud applications and the mainframe. Russ is the author of the US Patent around which the HostBridge “integration analytics” platform (HTAC) is based.



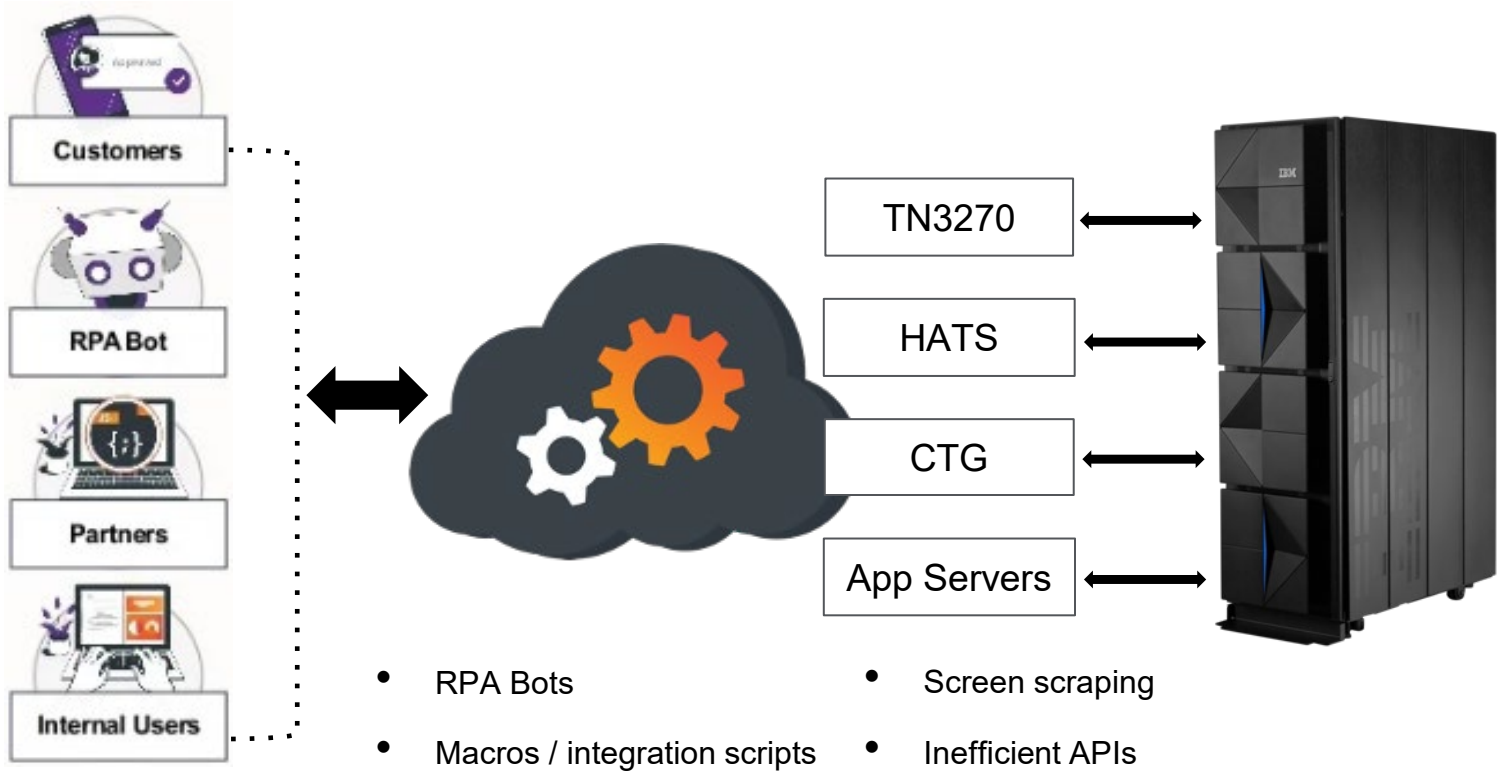
Greg Smith

Greg is a Software Engineer developing data analytics solutions for the HTAC platform. Greg is the primary developer, and customer engagement lead, for the HTAC “Macro Hunter” component. Greg works closely with clients to understand their requirements, generate work-flows, and design processes and dashboards to enable clients to find and fix their mainframe integration problems. Greg came to Broadcom via their acquisition of HostBridge Technology. Before joining HostBridge, Greg worked for 30+ years as a Lead System Engineer in the MIL-AERO industry.

BROADCOM SOFTWARE

Integration Analytics… Why?!

Our View: Inefficient Integrations Are a Significant Barrier to Modernization



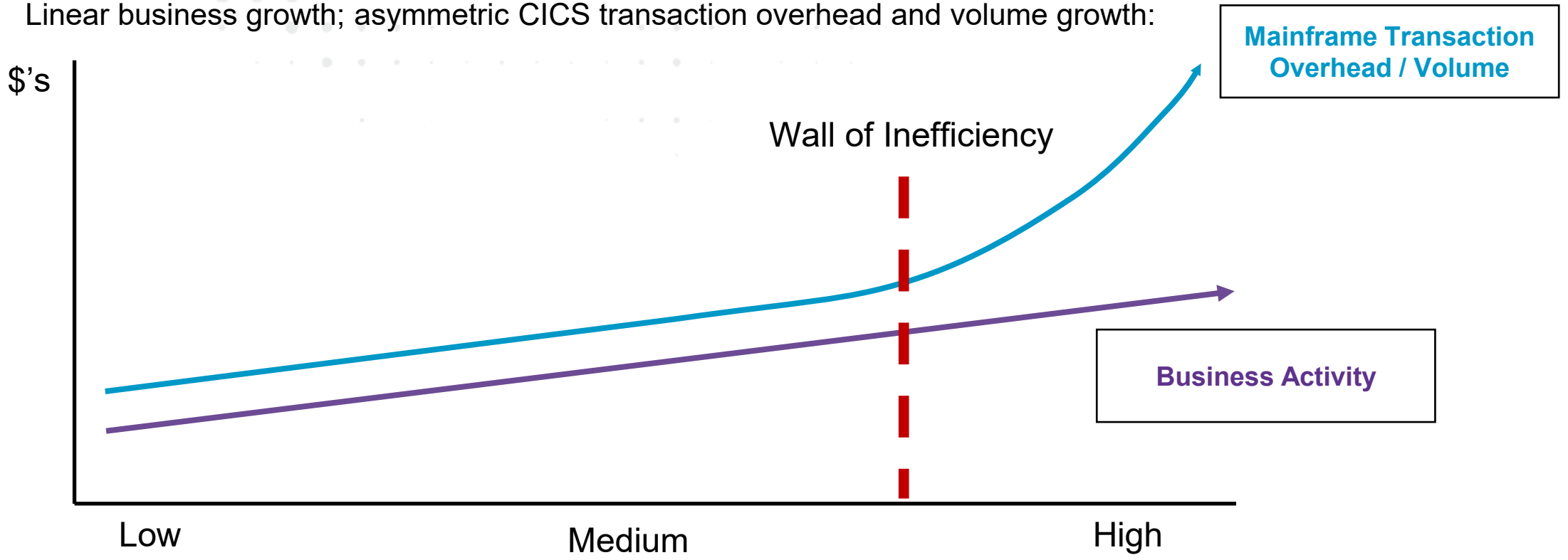
- RPA Bots
- Macros / integration scripts
- Screen scraping
- Inefficient APIs

- Inefficient integrations
  - Inject Latency
  - Degrade QOS
  - Add Transactional Load
  - Waste MIPS
  - Compromise ROI
  - Makes it difficult to modernize apps
  - Create bias against the mainframe

BROADCOM SOFTWARE

Inefficient Integration Surfaces in Various Ways

Linear business growth; asymmetric CICS transaction overhead and volume growth:



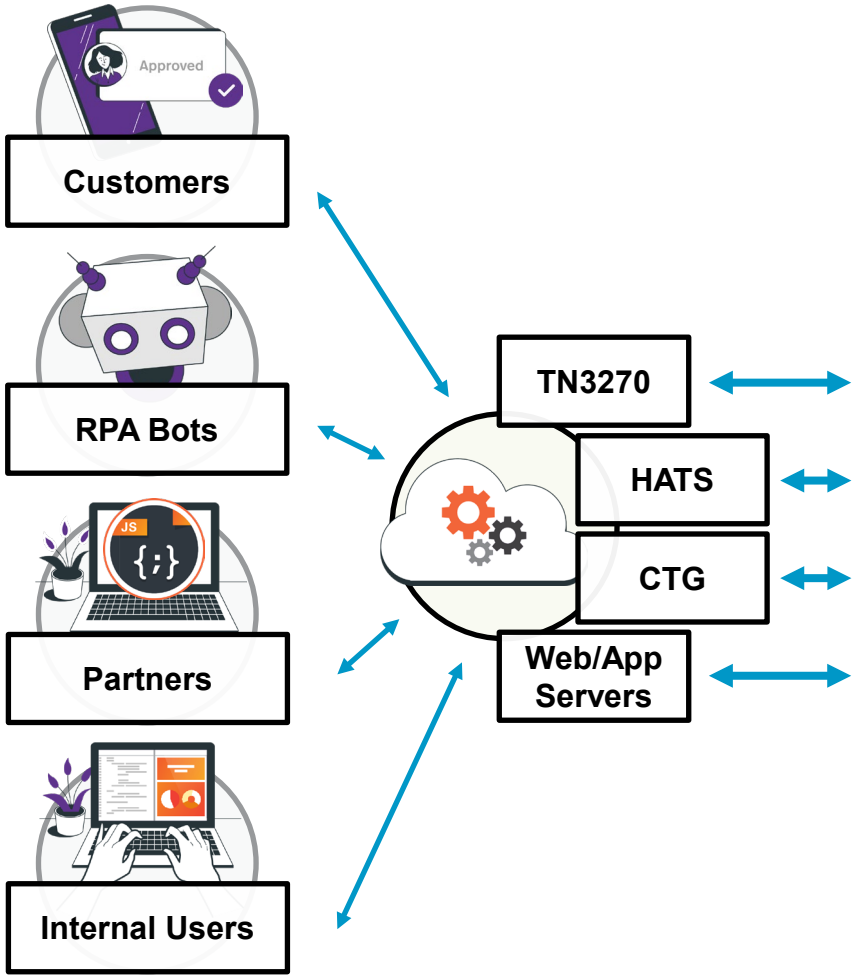
What's causing the surge? Scaling the business using inefficient mainframe integrations

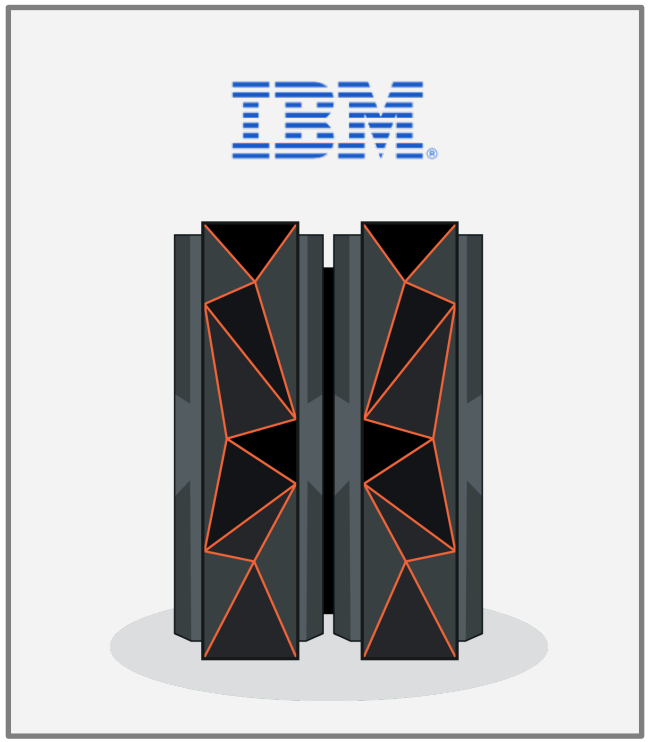
BROADCOM SOFTWARE

You Can’t Fix What You Can’t See

- HTAC is a diagnostic platform designed to find and analyze inefficient patterns of CICS integrations
  - We refer to this as “integration analytics”
- Objectives
  - Find common and wasteful patterns of integration
  - Quantify the impact
    - Latency / QOS
    - MIPS
  - Reveal the business purpose
    - Show the “DNA” of the integration (the sequence of transactions executed)
    - Allow application SMEs to understand the business purpose
  - Provide actionable intelligence
    - Highlight most impactful integrations (in terms of CPU consumed, latency injected, etc.)
    - You don’t need to “boil the ocean” (correct ALL issues) to achieve significant savings and operational gains
    - You just need to find and fix the most problematic cases

Integration Analytics: The Landscape



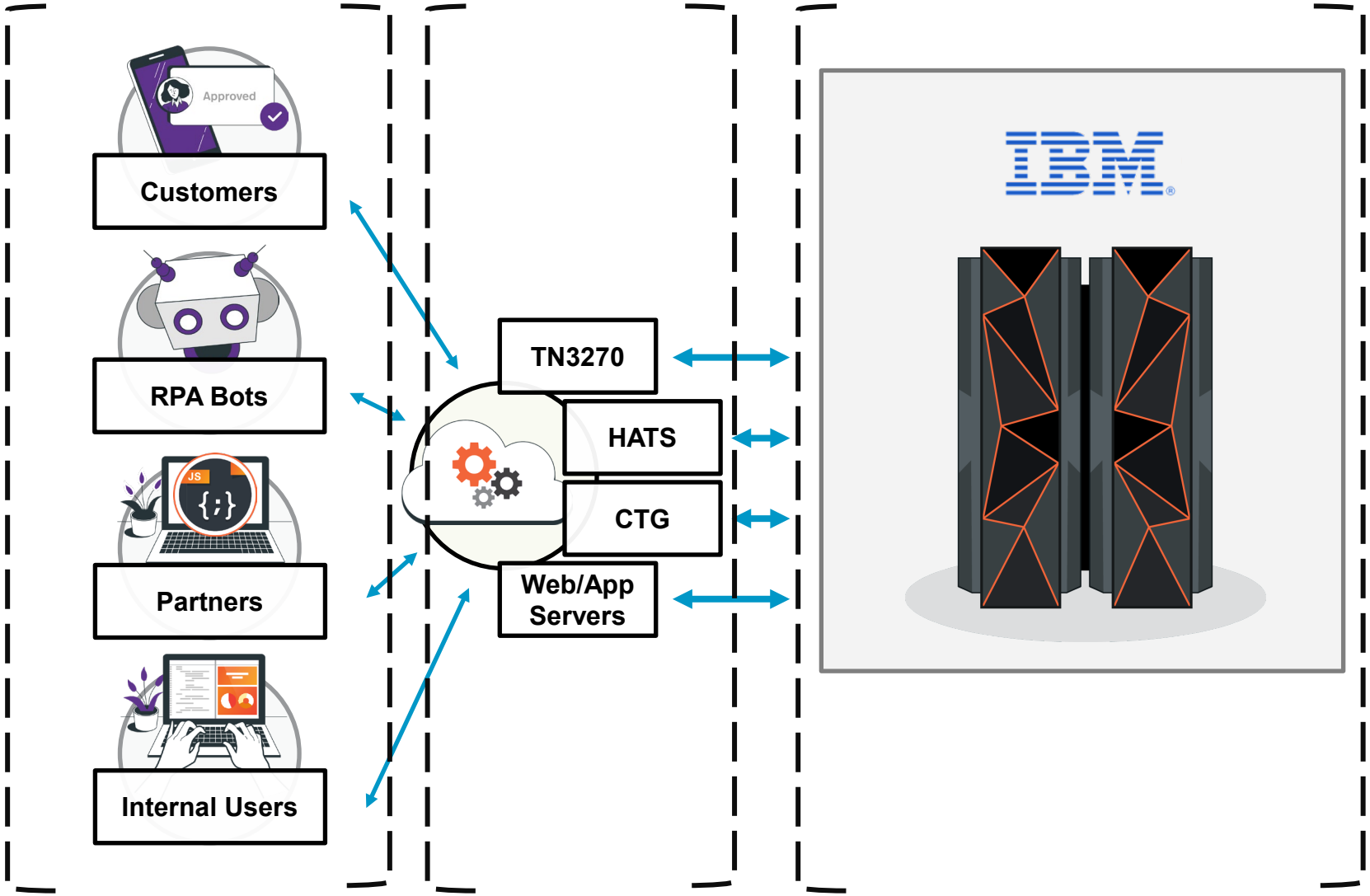


All organizations that use IBM mainframes operate in a hybrid environment.

Regardless of their specific configuration, they have thousands of end points driving work through multiple channels.

BROADCOM SOFTWARE

Integration Analytics: The Need



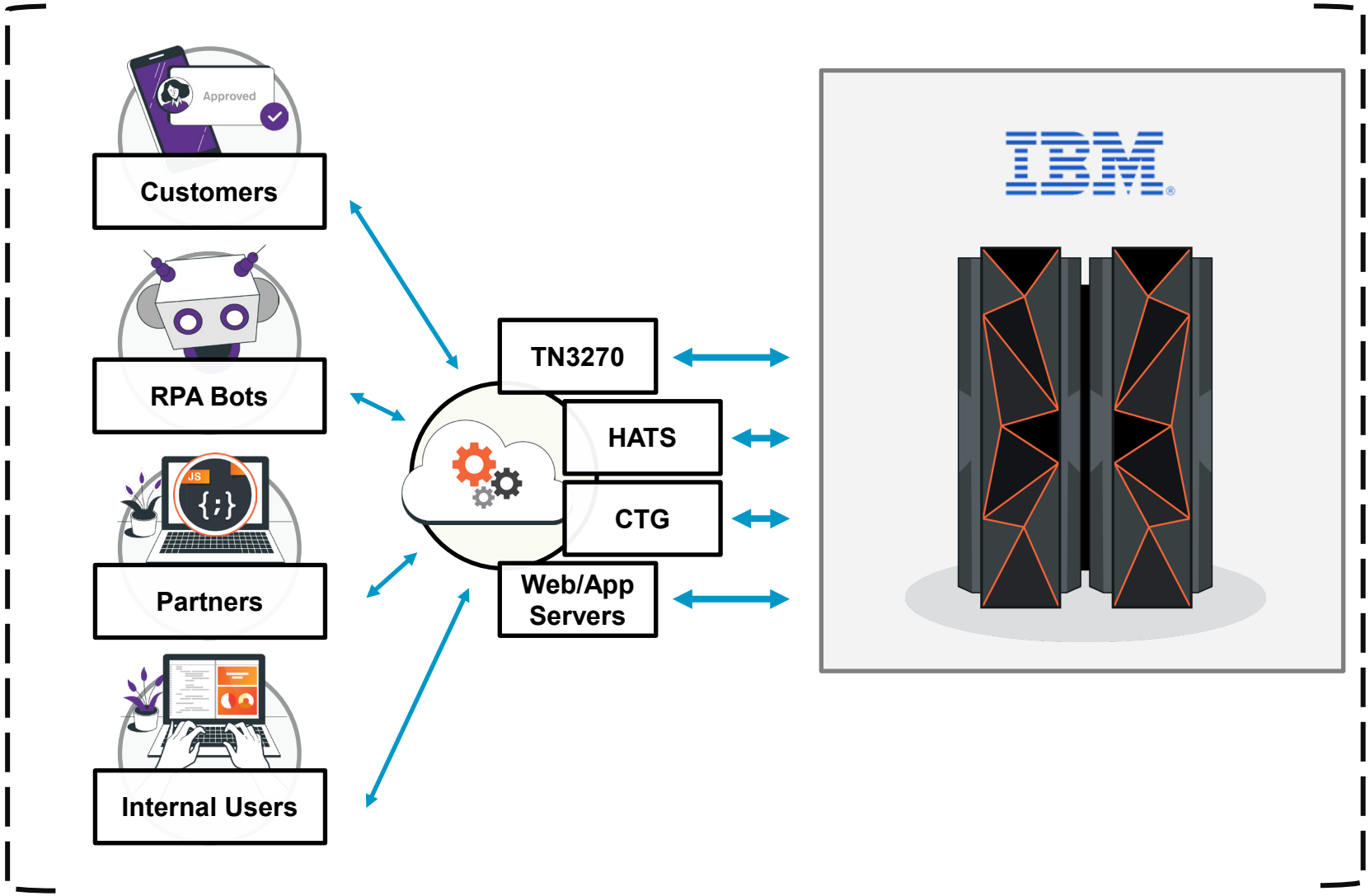
Most organizations have tools to evaluate the efficiency of each “silo” (within brackets). They can see the “trees” but not the “forest”.

- They cannot track mainframe load back to the origin.
- They cannot assess the business purpose and value.
- They cannot assess true cost.

Optimizing each silo does NOT ensure that the overall integration is efficient.

BROADCOM SOFTWARE

Integration Analytics: The Need (cont'd)

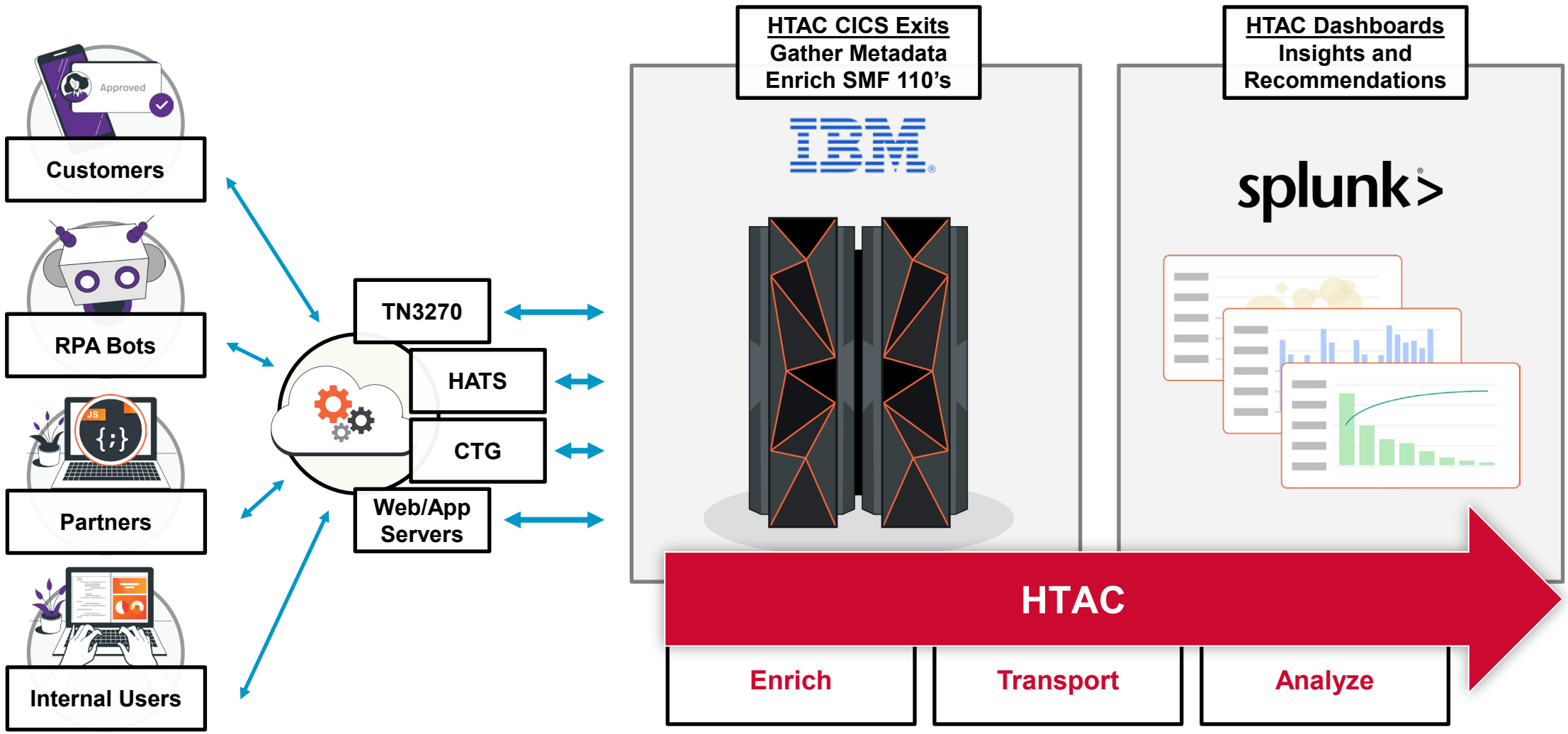


Integration Analytics allows you to see the "forest" not just the "trees" - and understand what is driving your mainframe workload.

- You can track transaction activity back to the origin.
- You can assess the business purpose and value.
- You can analyze the performance and cost of the integration.

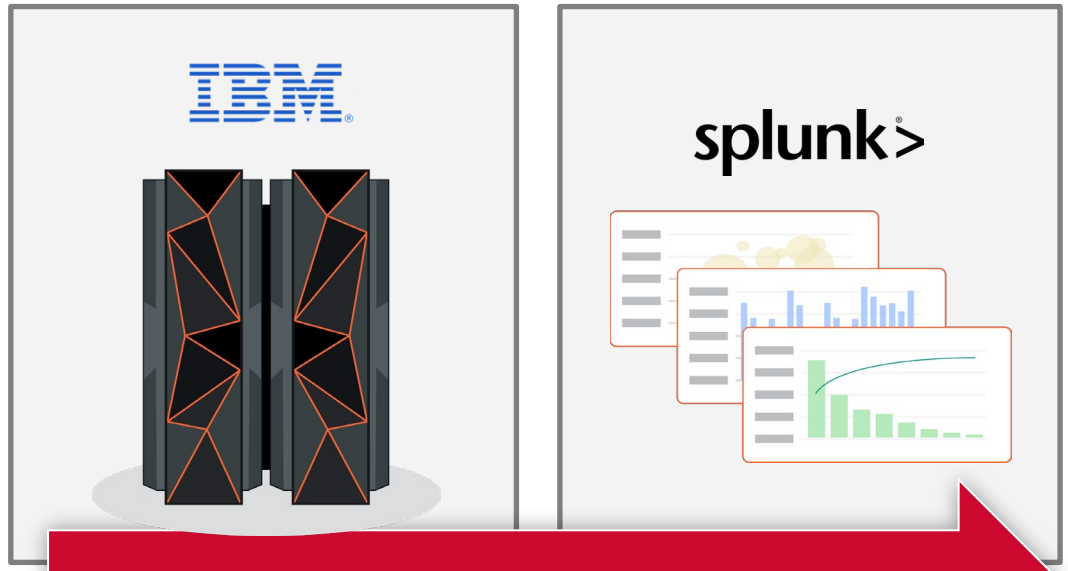
BROADCOM SOFTWARE

Integration Analytics: The Framework



BROADCOM SOFTWARE

HTAC Key Functions



CICS exits (or event driven programs) that extract metadata from targeted requests and enrich the CICS Monitor Data control block (which becomes the SMF 110).

Programs and processes tha: (a) transform CICS Monitor Data and/or SMF 110's into an HTAC-specific JSON document, and (b) send those JSON docs to Splunk as events (real time or batch).

Splunk-based code and processes that analyze the SMF110 events to detect inefficient integrations and visualize statistical results.

BROADCOM SOFTWARE

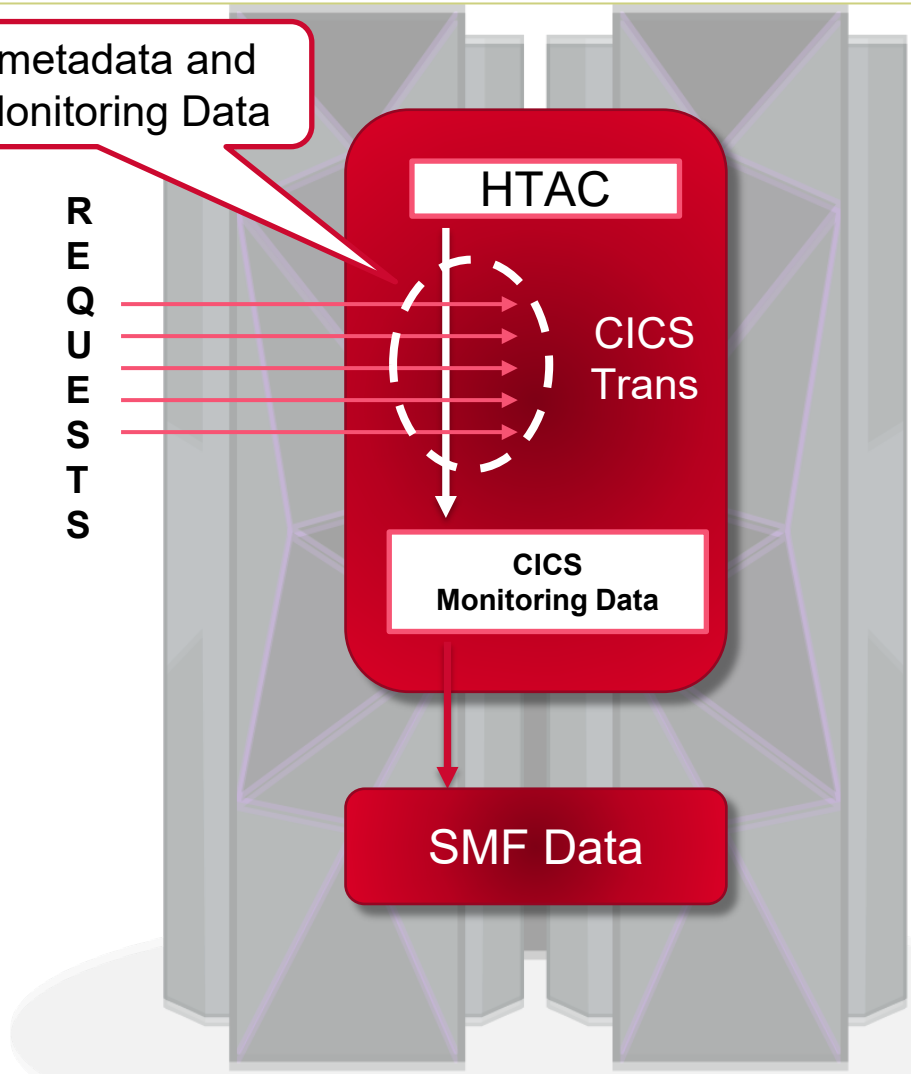
HTAC CICS Exits: Extract and Enrich

Software Objective

- Extract metadata from/about each targeted request
- Enrich standard CICS Monitor data in pre-existing “Origin Data” fields (e.g. OADID, OADATAn, OUSERCOR)
- We do not add fields to SMF 110 record or require changes to your MCTs
- Optional, but highly desirable for greatest insight

Software Implementation

- Low impact / very efficient
  - All code written in Assembly language
  - Code paths are negligible for cases not being monitored
- Designed to be run 7x24x365 or on-demand
- Implementation (depending on objective)
  - CICS Task Related User Exist (TRUE)
  - CICS Global User Exists (GLUE)
  - CICS Event Processing (EP) definitions/programs



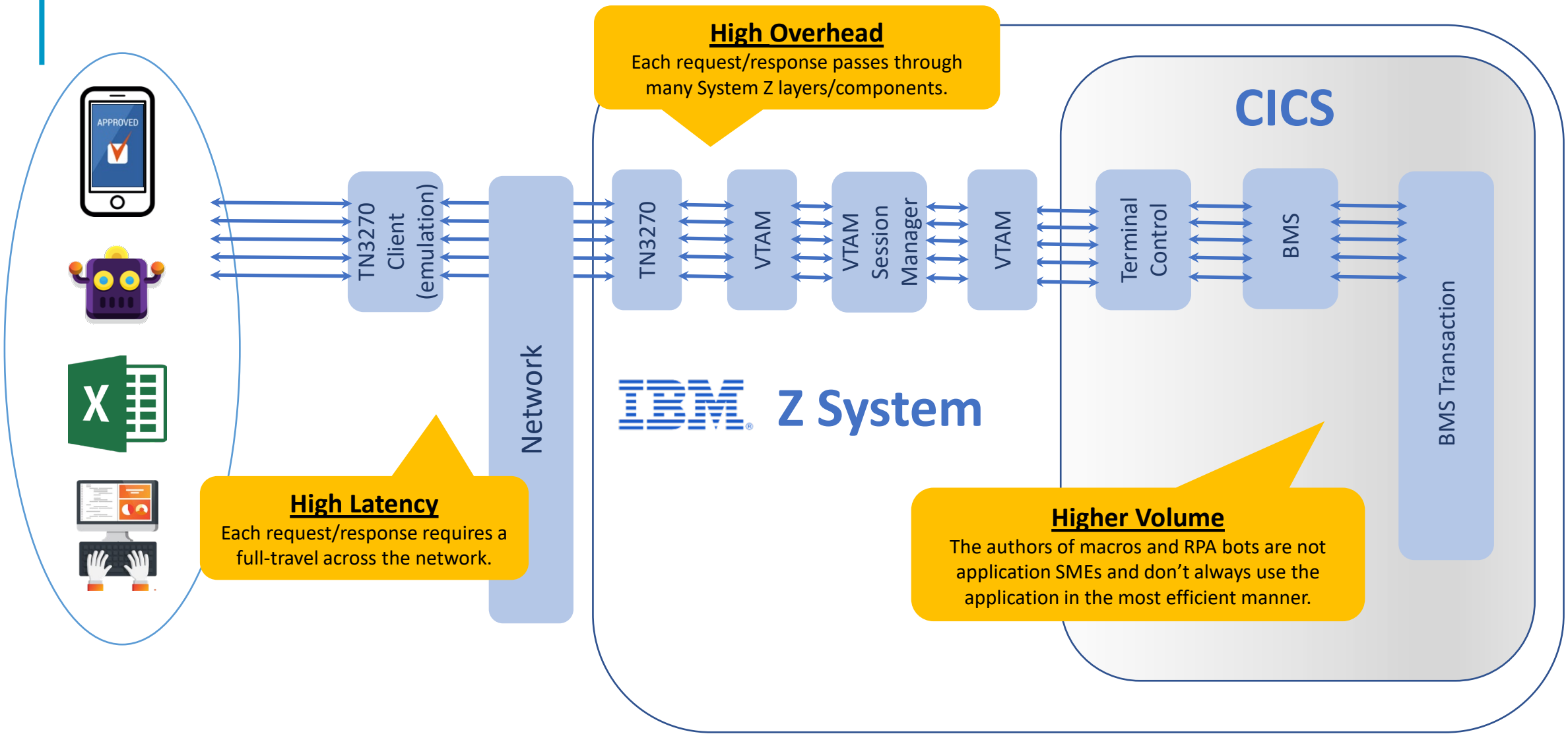
BROADCOM SOFTWARE

Overview of HTAC and Macro Hunter

What is “Macro Hunter”

- The HTAC framework is generic and can be applied to many different problems associated with mainframe integration
- However, based on early customer experiences, one style of inefficient integration was predominant:
  - ***Client, server or cloud-based programs/tools (off the mainframe) automating the execution of screen-oriented mainframe apps***
  - It doesn’t matter how old or new the tool may be, they are using Terminal Emulation and Screen Scraping to achieve integration
- Examples
  - Terminal Emulator scripts and macros (e.g., Rumba, Attachmate, PCOMM)
  - Excel macros using embedded VBA scripts
  - IBM HATS macros
  - UiPath RPA bots
- Some uses are harmless, but some are very harmful

Why Is Such Automation Problematic?



BROADCOM SOFTWARE

What is “Macro Hunter” (cont’d)

- To fully address this use case, we:
  - Enhanced the CICS exits to extract metadata germane to screen-oriented apps (e.g., AID key, BMS map name, program name).
  - Developed significant additional Splunk-based processes and code to detect these patterns of interaction, deduce their “DNA” (the steps in the automation), and assess their impact.
- Thus, “Macro Hunter” denotes the collection of HTAC components and features used to target this particular integration problem
  - Macro Hunter is included in HTAC
  - HTAC is single Broadcom SKU

Macro Hunter Terms and Concepts

- "Macro", "RPA", "Bot", "Script"
  - Used interchangeably to describe the program/tool that is automating the execution of screen-oriented CICS transactions.
  - Macro Hunter doesn't care about the automation source (Excel, VBA, UiPath, custom program, etc.)
  - Analytically, they all look the same (and have the same impact)

- "DNA"
  - The DNA of a macro/RPA is the sequence of steps it performs
  - DNA sequences can be long!

- "Signature"
  - The DNA of a macro/RPA is expressed as an MD5 hash value
  - Example: 3c640131b90fcc1940e6a97c6864a67e
  - For convenience we refer to a signature by it's first four characters (e.g., "3c64")
  - Thus, each DNA sequence has a unique identifier
  - This unique identifier allows us to "hunt" for the most common/problematic cases

BROADCOM SOFTWARE

Macro/RPA Analytics

rpaDuration	rpaTxDuration	Dur_Ratio	rpaCPUsec	rpaTotal	rpaTps	rpaStats	rpaDNA
00:22:30.297	00:01:33.189	14.5:1	00:00:27.541	10503	7.8	ORR=10503	ORR+ENTER*10503

Time between start of first transaction to end of last transaction. This RPA took 22 minutes and 30 seconds.

Sum of the duration of each individual transaction. The 10,503 transactions actually took 1 minute and 33 seconds to run.

Ratio of rpaDuration to rpaTxDuration. Indicates how much real time was potentially wasted due to latency and automation off the mainframe.

Sum of the CPU time of each individual transaction.

Total transactions executed as part of the RPA.

Rate (per second) which the RPA executed transactions.

Count of each transaction executed by the RPA.

The exact sequence of steps executed by the RPA.

BROADCOM SOFTWARE

The DNA Is The Treasure We Are Hunting For

The DNA includes sufficient detail to allow an app SME to understand *WHAT* the RPA is doing, and consider *HOW* the business requirement might be performed more efficiently.

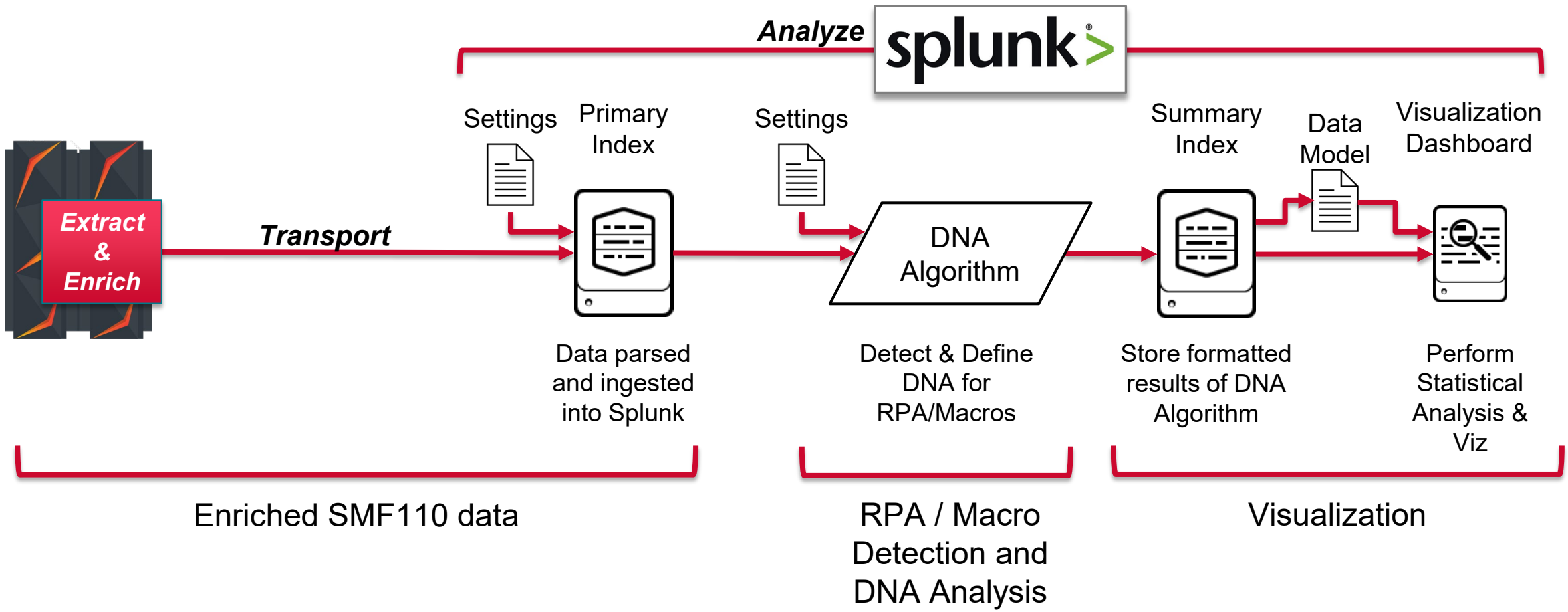
rpaTotal	rpaTps	rpaStats	rpaDNA
36	2.9	ICMA(CIA4)>CM10M03.CM10M03(CM10P003)=3 IROM(COMM)>RMRCM11.RMRCM11(RMRCP110)=3 PB1H(CIA4)=3 PBSE(CIA4)=5 PBSE(CIA4)>CG10M13.CG10M13(CG10P130)=3 PBSE(CIA4)>CM10M03.CM10M03(CM10P003)=7 =3	PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+ENTER>CG10M13.CG10M13(CG10P130) PB1H(CIA4)+PF21 SRTE+PF21 IROM(COMM)+PF21>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF18>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF2 SRTE+PF2 ICMA(CIA4)+PF2>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+CLEAR PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003)*2 PBSE(CIA4)+CLEAR PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+ENTER>CG10M13.CG10M13(CG10P130) PB1H(CIA4)+PF21 SRTE+PF21 IROM(COMM)+PF21>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF18>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF2 SRTE+PF2 ICMA(CIA4)+PF2>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+CLEAR PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+ENTER>CG10M13.CG10M13(CG10P130) PB1H(CIA4)+PF21 SRTE+PF21 IROM(COMM)+PF21>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF18>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF2 SRTE+PF2 ICMA(CIA4)+PF2>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+CLEAR PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003)*2 PBSE(CIA4)+CLEAR

- Transaction PBSE ran in AOR CIA4
- Due to an ENTER key
- The output of PBSE was BMS mapset/map CM10M03
- Program CM10P003 performed the EXEC CICS SEND MAP
- This same operation occurred twice (back-to-back)

BROADCOM SOFTWARE

HTAC and Macro Hunter: Behind the Scenes

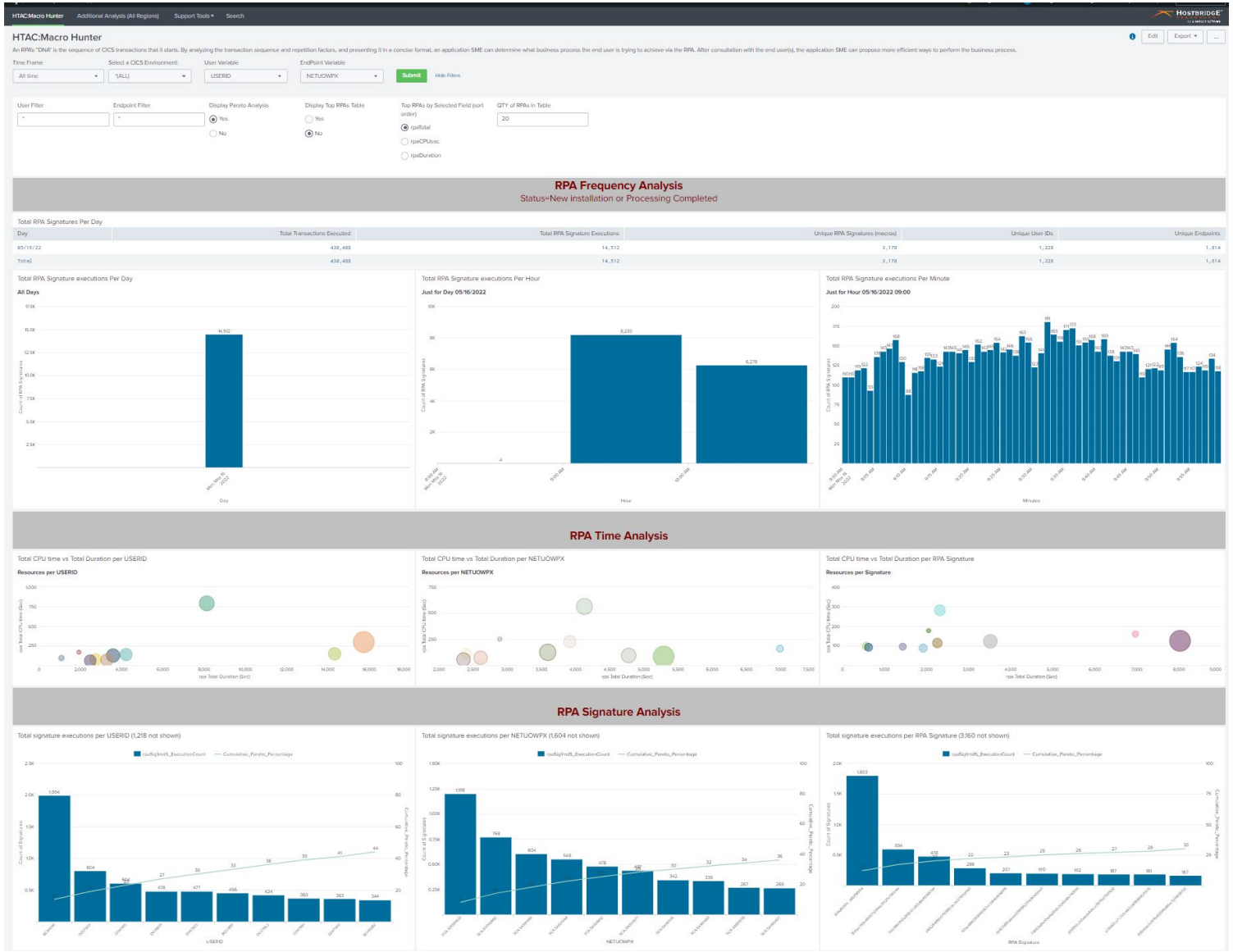
HTAC: Macro Hunter Architecture and Data Flow



BROADCOM SOFTWARE

HTAC Macro Hunter Dashboards

HTAC: Macro Hunter Dashboard (partial)



Menu
Filtering, Enable Visualizations

Day, Hour, Minute
RPA Time Distribution

Bubble Charts
CPU vs Duration

Pareto Charts
Top 10 RPAs per category

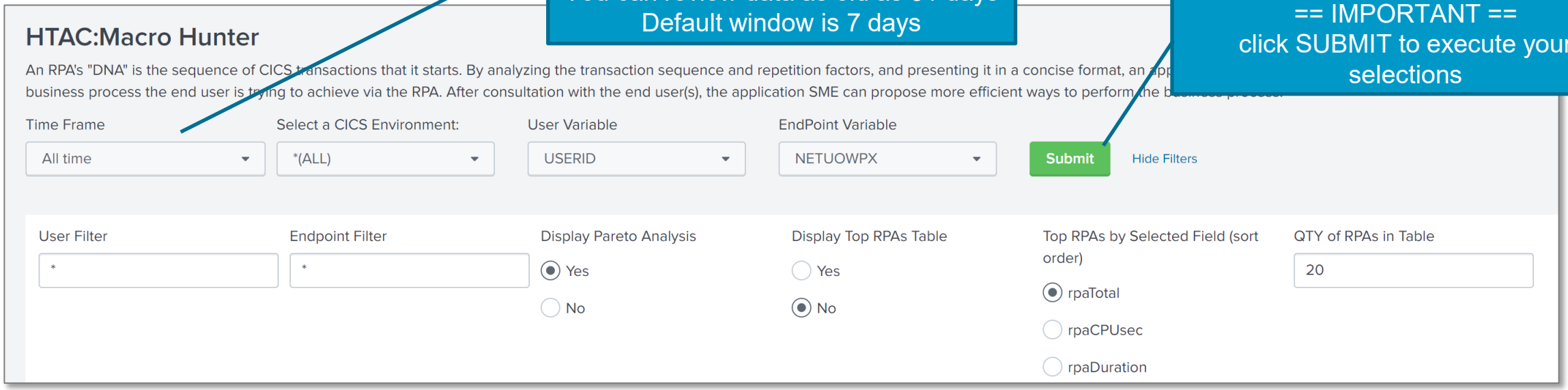
BROADCOM SOFTWARE

Menu Options

== IMPORTANT ==
This time frame is associated with the processed data itself.

You can review data as old as 31 days
Default window is 7 days

== IMPORTANT ==
click SUBMIT to execute your selections



- Time Frame
  - Pulldown list
  - Defaults to last 7 days
  - Can be user-defined
  - Filters processed data on selected time frame
  - All dashboard charts are relevant to this Time Frame
- Select a CICS Environment
  - Pulldown list
  - Defaults to “*” (ALL)
  - One environment at a time
  - Filters to selected CICS environment
  - Groupings of OAPPLIDs
- User Filter and Endpoint Filter
  - Text entry
  - Defaults to “*” (all)
  - Filters dataset to only the content of the associated field
- All other menu options
  - Provide display options
  - Recommend using defaults

BROADCOM SOFTWARE

Time-Based Drilldown: Day, Hour, Minute

1. Select a Day to display the Hours
2. Click on an Hour to display the Minutes
3. Click on a Minute to open another tab with data

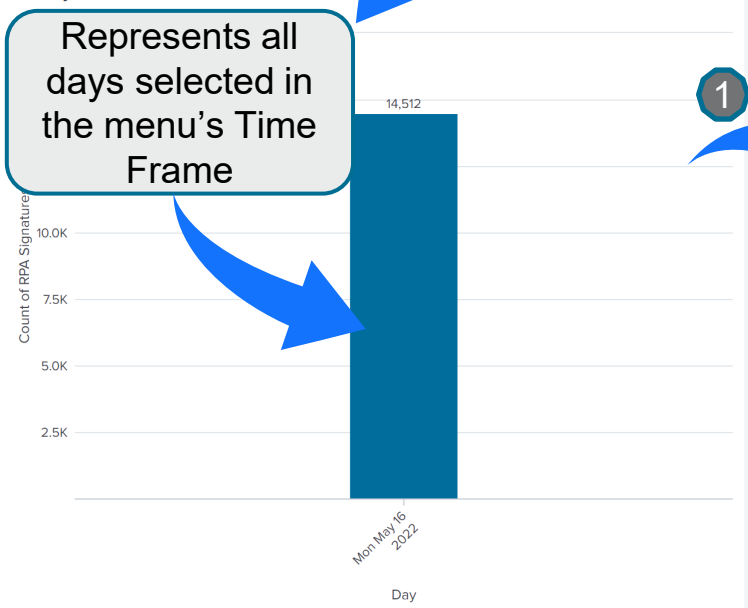
RPA Frequency Analysis

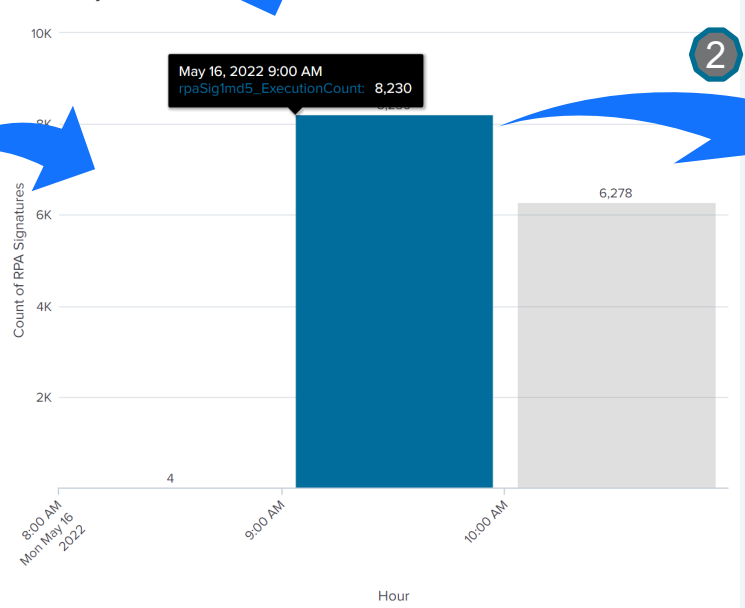
Status=New installation or Processing Completed

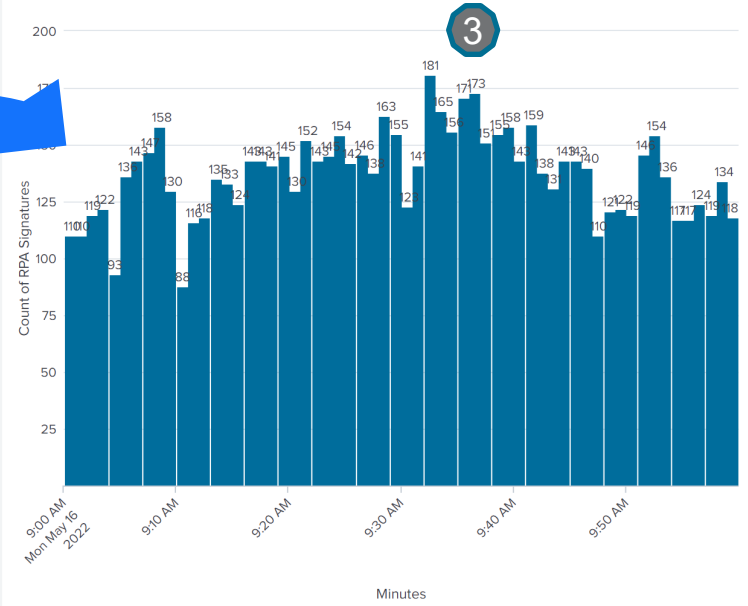
Total RPA Signatures Per Day

Day	Total Transactions Executed	Total RPA Signature Executions	Unique RPA Signatures (macros)	Unique User IDs	Unique Endpoints
05/16/22	430,488	14,512	3,170	1,228	1,614
Total	430,488	14,512	3,170	1,228	1,614

Represents all days selected in the menu's Time Frame







BROADCOM SOFTWARE

Drilldown from the Minute Panel

CICS Environment = *(ALL)

Total RPA Signature executions for this Minute

Format: 00:00:01.944908 ➔ Hr:Min:Sec:SubSec

Format: 2022-05-16 09:32:59.713 ➔ Year-Month-Day Hr:Min:Sec:SubSec

Can sort on any column

Minute is ...	09:32 (Hour:Min)
Total RPAs executed	181

from Minute Chart

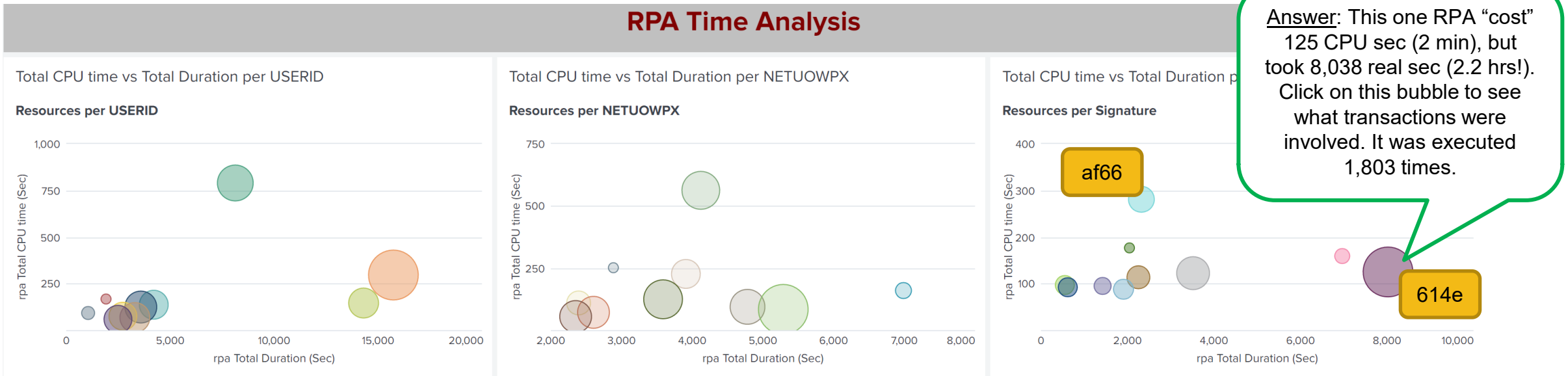
	NETUOWPX	OAPPLID	GAPPLID	cicsEnvir	USERID	rpaSTART	rpaSTOP	rpaDuration (HH:MM:SS:mSec)	rpaTxDuration (HH:MM:SS:mSec)	Dur_Ratio	rpaCPUsec (HH:MM:SS:mSec)	rpaTotal	rpaTps	rpaStats	rpaDNA
1	SCA.SD004665	A70MILL1	A70CIA4 A70COMMB A70MILL1	ScriptingTORs	BCHS582	2022-05-16 09:32:59.713	2022-05-16 09:33:12.200	00:00:12.486437	00:00:01.944908	6:1	00:00:00.229981	36	2.9	ICMA(CIA4)>CM10M03.CM10M03(CM10P003)=3 IROM(COMM)>RMRCM11.RMRCM11(RMRCP110)=3 PB1H(CIA4)=3 PBSE(CIA4)=5 PBSE(CIA4)>CG10M13.CG10M13(CG10P130)=3 PBSE(CIA4)>CM10M03.CM10M03(CM10P003)=7 RMRC(COMM)=3 RMRC(COMM)>RMRCM11.RMRCM11(RMRCP110)=3 SRTE=6	PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+ENTER>CG10M13.CG10M13(CG10P130) PB1H(CIA4)+PF21 SRTE+PF21 IROM(COMM)+PF21>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF18>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF2 SRTE+PF2 ICMA(CIA4)+PF2>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+CLEAR PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003)* PBSE(CIA4)+CLEAR PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+ENTER>CG10M13.CG10M13(CG10P130) PB1H(CIA4)+PF21 SRTE+PF21 IROM(COMM)+PF21>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF18>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF2 SRTE+PF2 ICMA(CIA4)+PF2>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+CLEAR PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+ENTER>CG10M13.CG10M13(CG10P130) PB1H(CIA4)+PF21 SRTE+PF21 IROM(COMM)+PF21>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF18>RMRCM11.RMRCM11(RMRCP110) RMRC(COMM)+PF2 SRTE+PF2 ICMA(CIA4)+PF2>CM10M03.CM10M03(CM10P003) PBSE(CIA4)+CLEAR PBSE(CIA4)+ENTER>CM10M03.CM10M03(CM10P003)* PBSE(CIA4)+CLEAR

=== Definitions ===

- rpaTotal = Total number of transactions within an automation
- rpaTps = The rate at which the Macro or RPA Bot executed CICS transactions (transactions per second)
- rpaStats = Alphabetized list of mainframe transactions and the number of times each transaction was executed for that RPA; This list does NOT imply sequence.
  - Ex: PB1H(CIA4)=3 means the PB1H command was executed 3 times for this RPA within AOR CIA4.
- rpaDNA = sequential listing of mainframe transactions and function keys comprising this RPA; Detailed sequence of CICS transactions executed as part of the automation sequence

BROADCOM SOFTWARE

Bubble Charts - CPU Execution Time vs RPA Duration

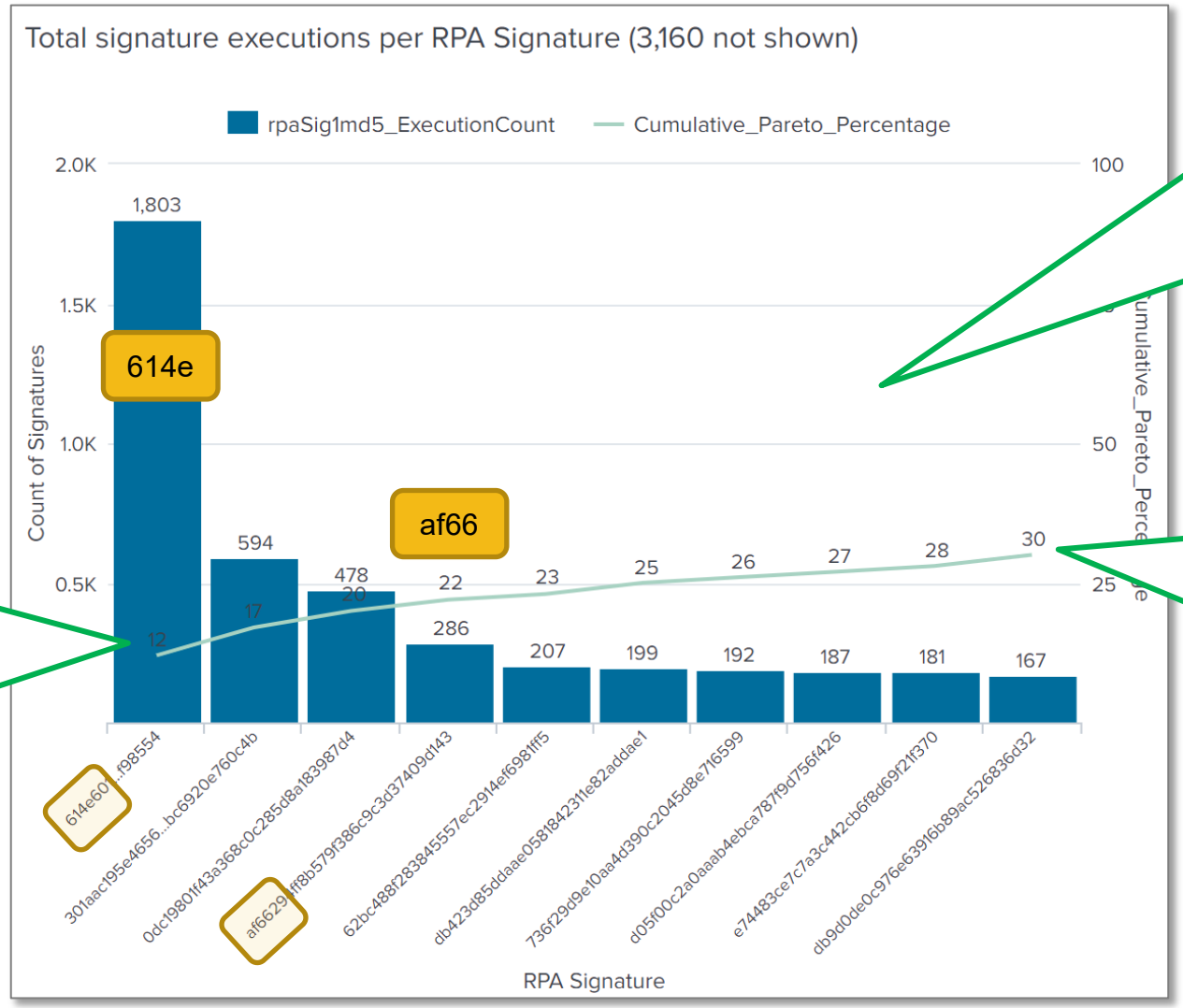


- USERID
  - **Who** executed the RPAs?
- NETUOWPX
  - **Where** did the RPA execution come from?
  - CLIPADDR is another option for this endpoint category
- RPA Signature / Macro
  - **What** RPA was executed?
  - The DNA of each RPA is deduced
  - RPAs with similar DNA have a common "signature"
  - Each signature expressed as an md5 hash value

BROADCOM SOFTWARE

Pareto – Total RPA executions

Question: Where should we focus our attention for RPA optimization and remediation?



Answer: In terms of total RPA executions, these are the top RPA signatures.

Focus optimization and remediation efforts here.

This single RPA represents 12% of ALL RPA executions within the selection criteria from the menu.

This # of executions is 3x that of its nearest RPA.

These 10 RPAs represent the top 30% of ALL RPA executions within the selection criteria from the menu.

BROADCOM SOFTWARE

Pareto – Total CPU Time per RPA

Question: Where should we focus our attention for RPA optimization and remediation?

Answer: In terms of total CPU usage, these are the top RPA signatures.

Focus optimization and remediation efforts here.

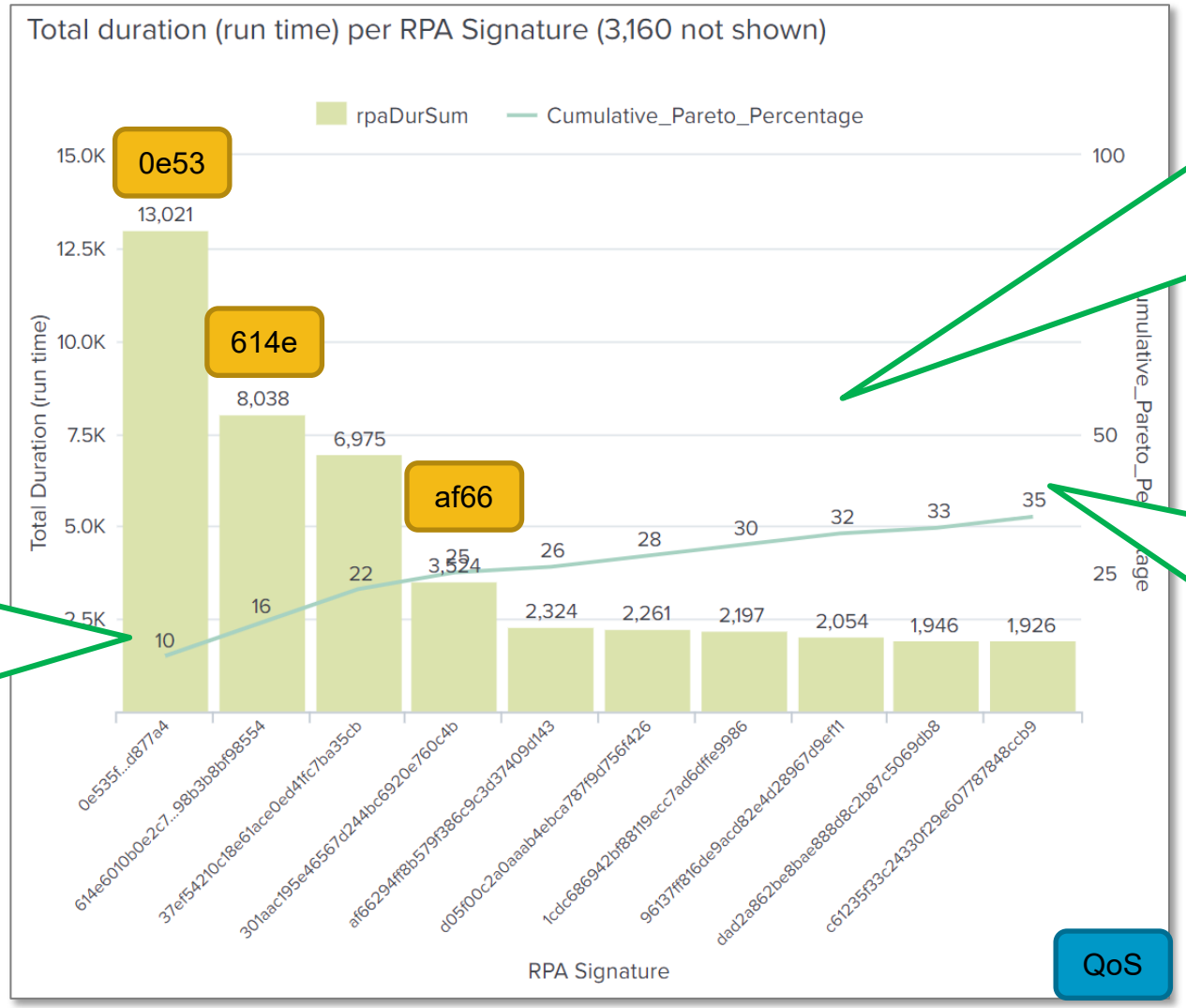
This single RPA represents 9% of ALL RPA CPU time within the selection criteria from the menu.

These 10 RPAs represent the top 42% of ALL RPA CPU execution times within the selection criteria from the menu.

BROADCOM SOFTWARE

Pareto – Total Duration per RPA

Question: Where should we focus our attention for RPA optimization and remediation?



Answer: In terms of total Duration, these are the top RPA signatures.

Focus optimization and remediation efforts here.

This single RPA represents 10% of ALL RPA Duration times within the selection criteria from the menu.

These 10 RPAs represent the top 35% of ALL RPA Duration times within the selection criteria from the menu.

BROADCOM SOFTWARE

Live Demo

Bad Bot Analysis

Discovering Bad Macros/Bots

DIMS+CLEAR>TERMMSM.TERMMA4
DIMS+ENTER>TERMMSM.TERMMA2
DIMS+ENTER>TERMMSM.TERMMA3*2
DIMS+CLEAR>TERMMSM.TERMMA4*2
DIMS+ENTER>PARTMSM.PARTMA1
DIMS+CLEAR>TERMMSM.TERMMA4*2
DIMS+ENTER>CON1MSM.CON1MA2
DIMS+CLEAR>TERMMSM.TERMMA4*2
DIMS+ENTER>VN01MSM.VN01MA1
DIMS+ENTER>VN02MSM.VN02MA1
DIMS+CLEAR>TERMMSM.TERMMA4*2
DIMS+ENTER>VN01MSM.VN01MA1
DIMS+CLEAR>TERMMSM.TERMMA4

This is the most commonly executed RPA sequence.

It accounts for more real time (latency) and CPU time than any other sequence.

BROADCOM SOFTWARE

Discovering Bad Macros/Bots

DIMS+CLEAR>TERMMSM.TERMMA4
DIMS+ENTER>TERMMSM.TERMMA2
DIMS+ENTER>TERMMSM.TERMMA3*2
DIMS+CLEAR>TERMMSM.TERMMA4*2
DIMS+ENTER>PARTMSM.PARTMA1
DIMS+CLEAR>TERMMSM.TERMMA4*2
DIMS+ENTER>CON1MSM.CON1MA2
DIMS+CLEAR>TERMMSM.TERMMA4*2
DIMS+ENTER>VN01MSM.VN01MA1
DIMS+ENTER>VN02MSM.VN02MA1
DIMS+CLEAR>TERMMSM.TERMMA4*2
DIMS+ENTER>VN01MSM.VN01MA1
DIMS+CLEAR>TERMMSM.TERMMA4

Sometimes it takes a curious human to spot a bad macro/bot!

Questions:

- Why is the CLEAR key used so frequently?
- And why is it used back-to-back?

BROADCOM SOFTWARE

Discovering Bad Macros/Bots

This CICS transaction sequence is automated via a Web UI/RPA platform (IBM HATS). The macro/bot that HTAC detected is implemented by a "main" HATS macro which invokes 5 smaller "subroutine" macros.

Each "subroutine" macro was written to be a "good macro" and defend against "bad macros":

- First op: CLEARs screen putting CICS app in known state
- Last op: CLEARs screen leaving CICS app in known state
- Even if the macro only runs a single transaction
- The macro authors did this with the best of intentions

However, the **unintended consequences** were significant:

- The application SME determined that NONE of the CLEAR keys were required to manage the application state
- Thus, 50% of ALL transactions driven by HATS had no functional or business value
- These macros wasted mainframe CPU time and degraded end user response time

Macro 1:

DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)
DIMS+ENTER>TERMMSM.TERMMA2(TERMMO)
DIMS+ENTER>TERMMSM.TERMMA3(TERMDI)*2
DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)

Macro 2:

DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)
DIMS+ENTER>PARTMSM.PARTMA1(PARTDI)
DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)

Macro 3:

DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)
DIMS+ENTER>CON1MSM.CON1MA2(CON1DI)
DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)

Macro 4:

DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)
DIMS+ENTER>VN01MSM.VN01MA1(VN01DI)
DIMS+ENTER>VN02MSM.VN02MA1(VN02DI)
DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)

Macro 5:

DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)
DIMS+ENTER>VN01MSM.VN01MA1(VN01DI)
DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)

Discovering Bad Macros/Bots

This CICS transactior
UI/RPA platform (IBM
detected is implemen
5 smaller “subroutine”

Each “subroutine” ma
defend against “bad n

- First op: CLEARs s
- Last op: CLEARs s
- Even if the macro
- The macro authors

However, the **uninter**

- The application SM
  keys were required
- Thus, <u>50% of ALL t
  functional or busine</u>
- These macros wasted mainframe CPU time and degraded end user response time

Macro 1:

MMSM.TERMMA4(TERMCT)
MMSM.TERMMA2(TERMMO)
MMSM.TERMMA3(TERMDI)*2
MMSM.TERMMA4(TERMCT)

MMSM.TERMMA4(TERMCT)
TMSM.PARTMA1(PARTDI)
MMSM.TERMMA4(TERMCT)

MMSM.TERMMA4(TERMCT)
1MSM.CON1MA2(CON1DI)
MMSM.TERMMA4(TERMCT)

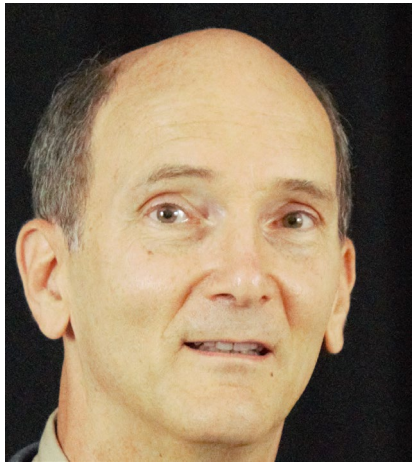
MMSM.TERMMA4(TERMCT)
MSM.VN01MA1(VN01DI)
2MSM.VN02MA1(VN02DI)
MMSM.TERMMA4(TERMCT)

DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)
DIMS+ENTER>VN01MSM.VN01MA1(VN01DI)
DIMS+CLEAR>TERMMSM.TERMMA4(TERMCT)

SOFTWARE

Other Use-Cases? Definitely!

- The HTAC framework is generic and intended to be applied to many different problems associated with mainframe integration
- Other use-cases have already been developed/deployed
  - Analysis of requests/responses between a server-based app and CICS transactions via a vendor specific socket protocol
  - Analysis of http requests/responses between a .Net server app and CICS
  - Extracting MQ Correlation ID and adding it to CICS Monitor Data (i.e., SMF 110) to enable end-to-end analytics
- Others are being evaluated: z/OS Connect
  - z/OS Connect creates an API around a single CICS program
  - If a business process requires the execution of multiple programs, then multiple API calls must be performed (typically via non-mainframe orchestration or automation)
  - This can devolve into a costly and inefficient pattern of integration
- What are YOUR integration challenges?



Russ Teubner
Russ.Teubner@broadcom.com

Greg Smith
Greg.Smith@broadcom.com

Q/A ?

BROADCOM SOFTWARE

Thank you

BROADCOM SOFTWARE