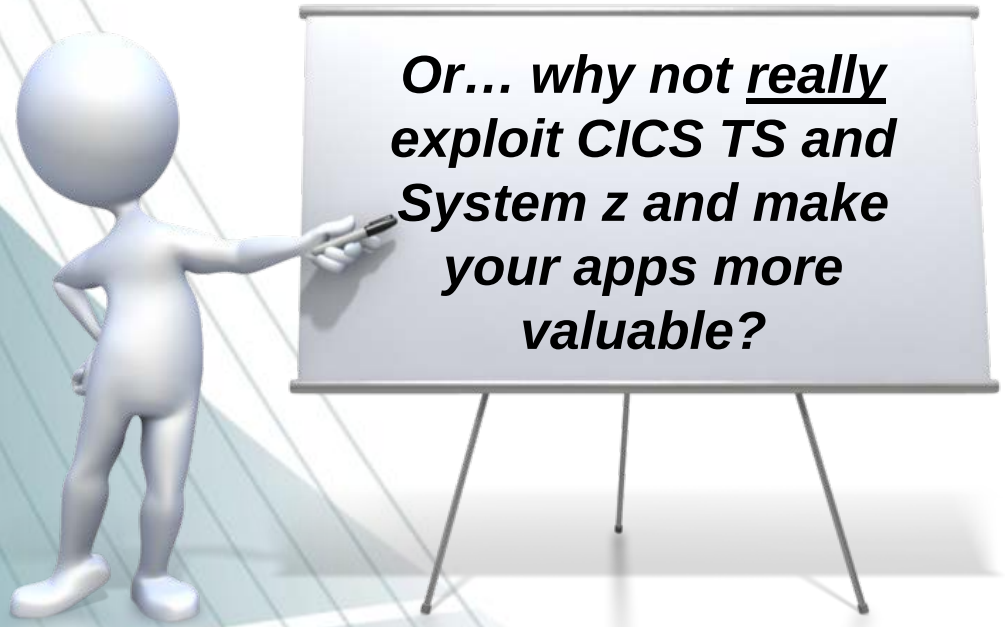
#SHAREorg

SHARE

CICS Integration & Optimization: Tales from the Trenches



Russ Teubner

HostBridge Technology

russ@hostbridge.com

Business Partner

Ready for Rational software

SHARE in Boston

Abstract

CICS users are loyal to their apps – and for good reason! However, they also need to integrate these same applications with an ever widening array of web, cloud and mobile resources. If that weren't enough, every year they are under pressure to support new workload and reduce the cost of ownership. That's a tall order.

Fortunately, IBM continues to deliver new versions of CICS that focus on operational efficiency and service agility. ISVs like HostBridge build upon these capabilities to help customers save time, reduce cost or generate revenue.

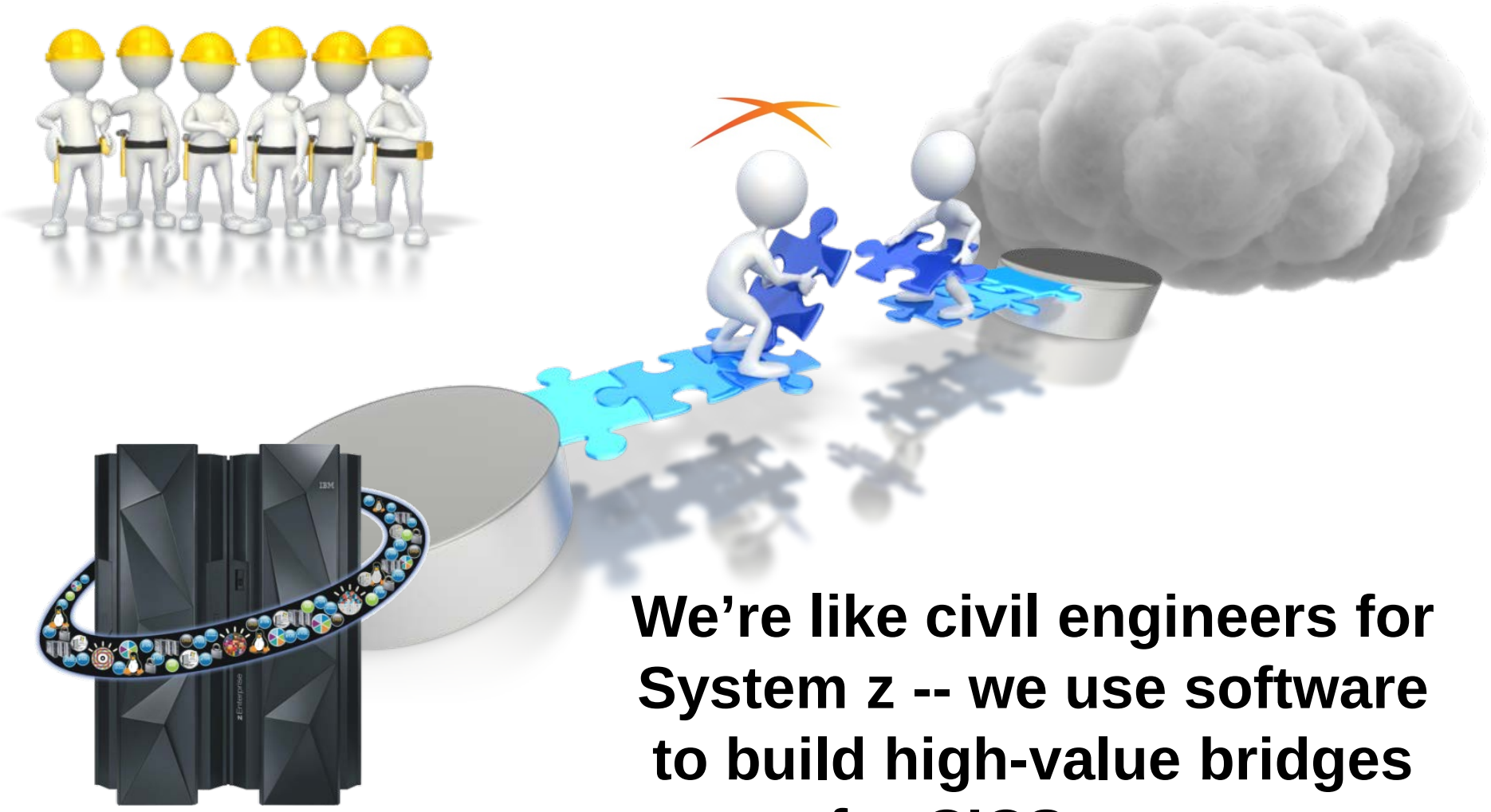
This session will highlight how HostBridge is exploiting CICS TS 5.1 and describe tactics that customers can use to enhance the value of their existing CICS investments (and lower their cost).

Who Is HostBridge?

- **CICS integration software company**
  - Founded in 2000 to exploit CICS TS and invent a new breed of integration products
  - We create and deliver software products that help CICS customers save time, cut costs and make money
- **Serving large organizations worldwide**
  - NISSAN, Navy Federal Credit Union, Wells Fargo, Edward Jones, Harland Clarke, PACCAR, Aegon UK, State of AZ, NYC Department of Education, City/County of San Francisco, Los Angeles County
- **Strong Technology Partnerships**
  - Strong working relationships with IBM System z, zOS, LE and CICS product groups

Business Partner IBM
Ready for Rational software

Stated Differently…



We’re like civil engineers for System z -- we use software to build high-value bridges for CICS apps.

Our Integration Principles

Principle #1: Integration is a means to an end. It must create immediate business value AND facilitate future strategic objectives

Principle #2: The integration layer should be agnostic to, yet exploit, the existing infrastructure.

zOS
TCP/IP
CICS HTTP
Web Services
Java Web Server
DataPower
ECI / EXCI
WAS
WOLA
ESB
MQ

Principle #3: The integration layer must be designed for performance, fidelity and future leverage.

CICS TS Exploitation

HostBridge Integration Engine

System z Exploitation

Principle #4: The integration layer must support all types of existing high-value apps and data.

Business Logic

Data

Visual Transactions

Principle #5: Integration workload should exploit the modern System z architecture to minimize cost and achieve high performance with enterprise class scalability. Existing high-value apps can consume higher-cost MIPS; new and/or integration workloads should exploit lower-cost MIPS.

CICS TS Environment

zAAP Work-load

GP Work-load

zIIP Work-load

Other System z Environments

zAAP Work-load

GP Work-load

zIIP Work-load

HostBridge System z Services

CICS Services

LE Services

Other Services

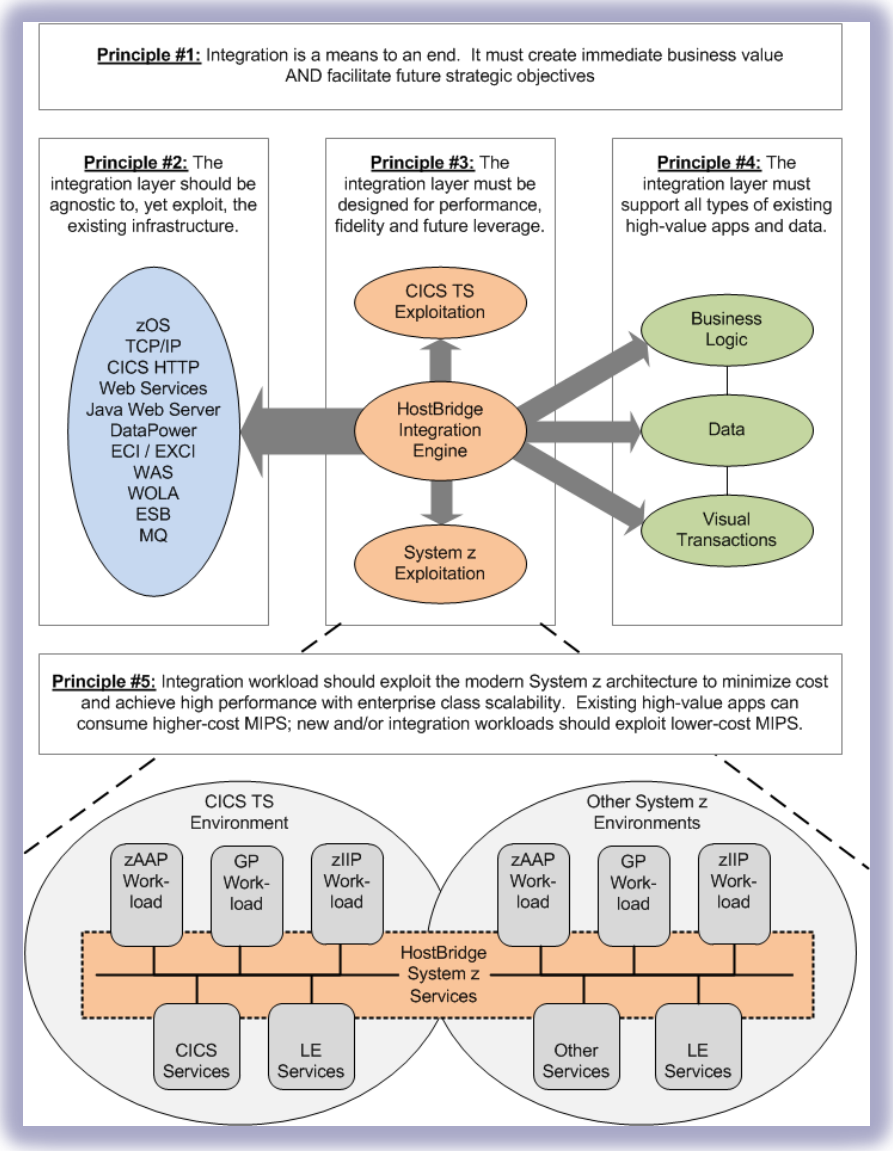
LE Services

Net Net?

- **We work to…**
  - **Integrate CICS apps with distributed apps in unique ways**
  - **Migrate workload to CICS in a cost effective manner**
  - **Lower the cost of ownership of CICS apps**
- **We obsess over…**
  - **Using high-value (GP) MIPS for high-value apps**
  - **Running everything else on specialty engines**
- **We write code to…**
  - **Exploit the modern CICS and System z environments**

Thus… Our Perspective

Whether your business objective is to reduce costs or generate revenue, HOW you approach integration matters.



CICS TS 5.1 – IBM Highlights

- **Improve Operational Efficiency**
  - Greater capacity
  - Managed operations
  - Increased availability
  - Deeper insight
- **Enhance Service Agility**
  - First-class applications
  - First-class platforms
  - Modern interfaces
  - Foundational enhancements

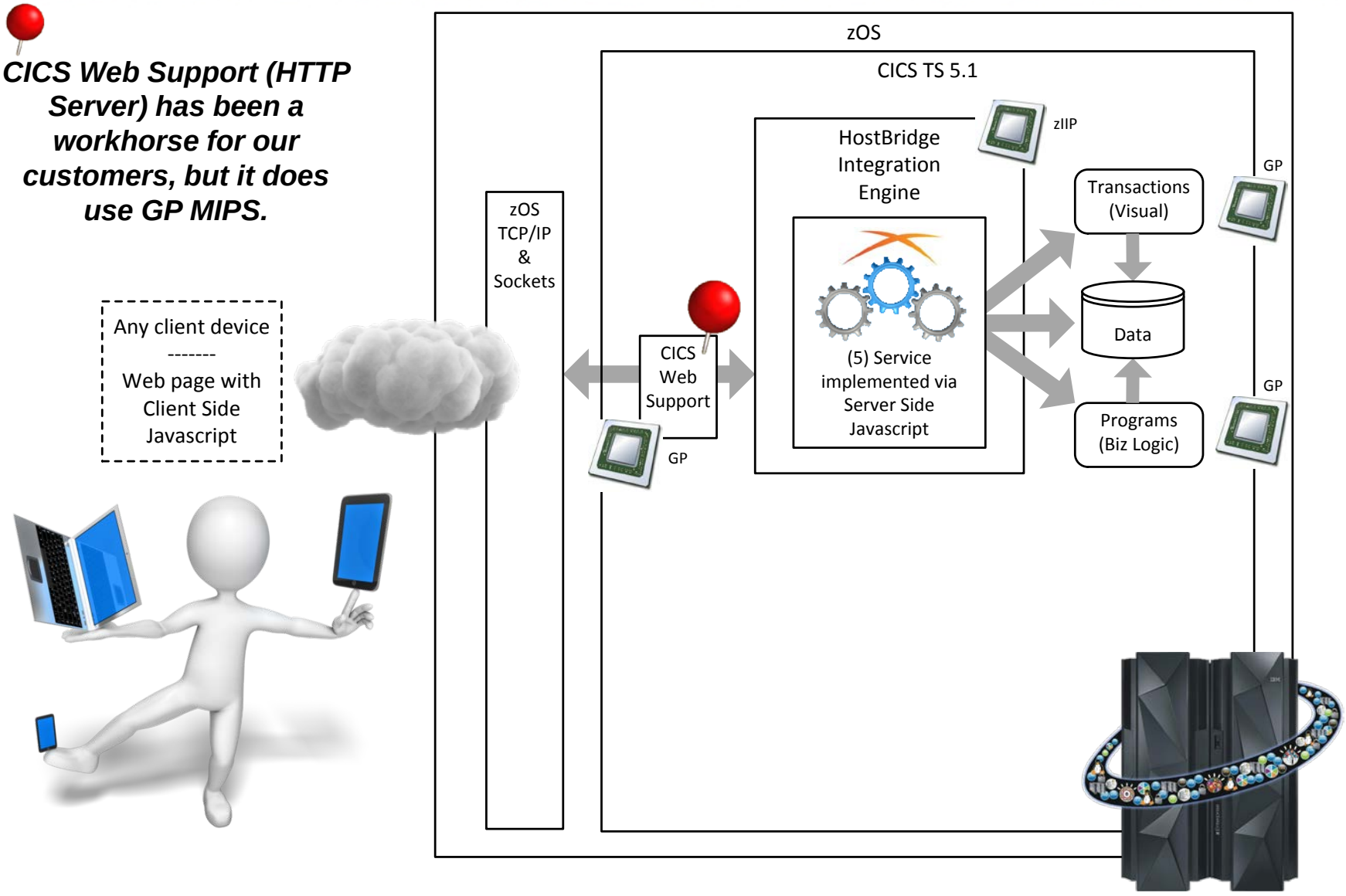
CICS TS 5.1 – Our Focus

- **OTE Enhancements and 64-bit Storage**
  - **We are a BIG user of CICS OTE**
  - **Making more EXEC CICS commands THREADSAFE is a very good thing**
  - **Greater use of 64-bit storage by CICS means greater availability of 31-bit storage for everyone else**
  - **Combined, these factors mean our customers can increase single-region load/parallelism further**

- **Java and WAS Liberty Profile**
  - **The significant Java improvements caught our attention**
  - **We can now consider using Java as a component in building CICS integration solutions that scale**
  - **The Web Container makes it easy to deploy Java™ servlets and Java Server Pages (JSPs)**
  - **The fact that CICS relies on the WebSphere® Application Server Liberty Profile means it's not going anywhere**

Typical Pre-CICS TS 5.1 Config

CICS Web Support (HTTP Server) has been a workhorse for our customers, but it does use GP MIPS.



Java Web Container Config

Communications handled via Java Web Container

Any client device

Web page with Client Side Javascript

zOS

CICS TS 5.1

zIIP

HostBridge Integration Engine

GP

Transactions (Visual)

zOS TCP/IP & Sockets

CICS Web Support

(5) Service implemented via Server Side Javascript

Data

GP

Programs (Biz Logic)

(4) LINK to HBJS Engine

(6) Return Response to JSP

(7) XML or JSON Returned

(3) Post XML or JSON to JSP

zAAP

JSP

Web Page

(2) Web Page Returned

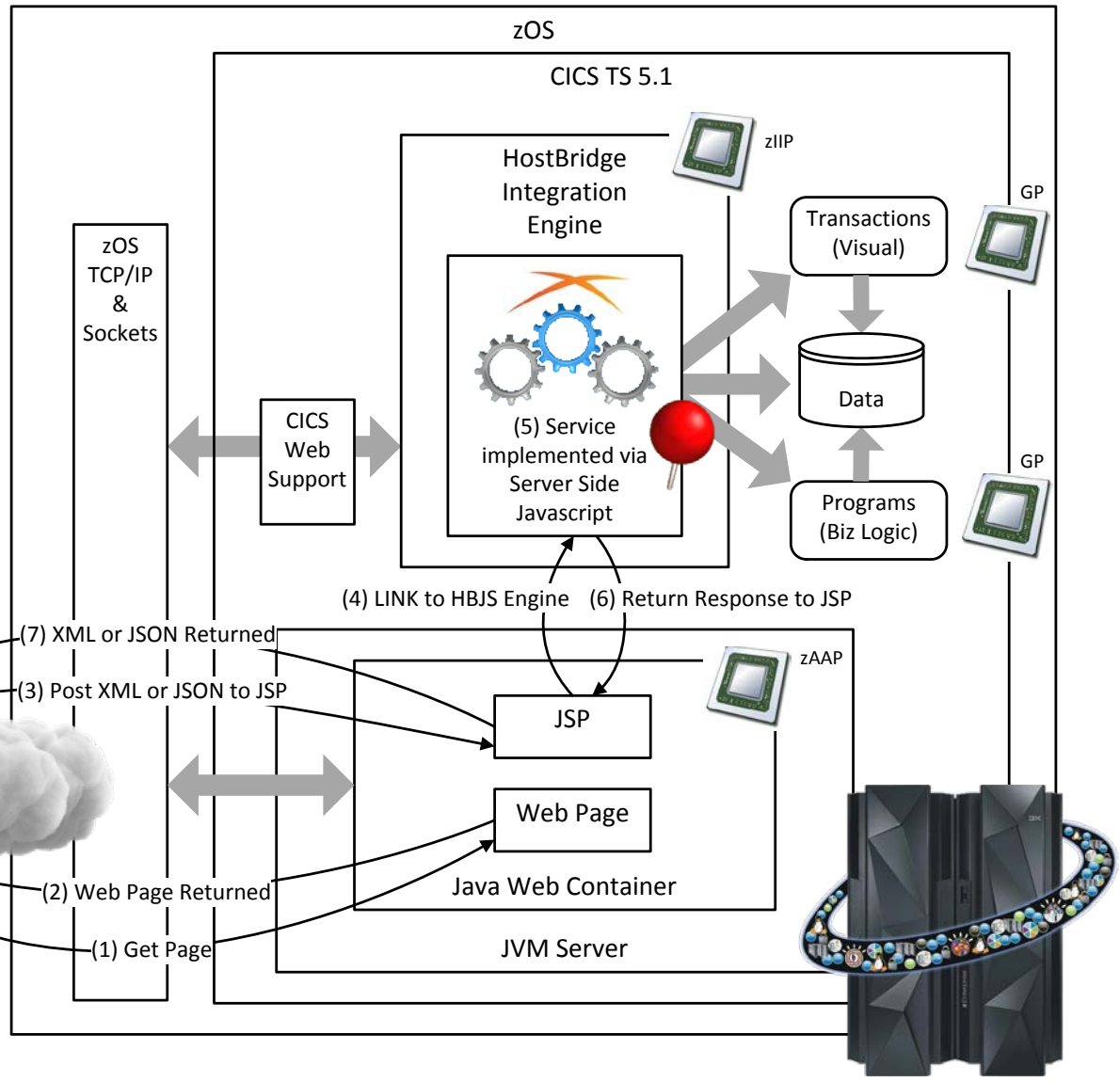
(1) Get Page

Java Web Container

JVM Server

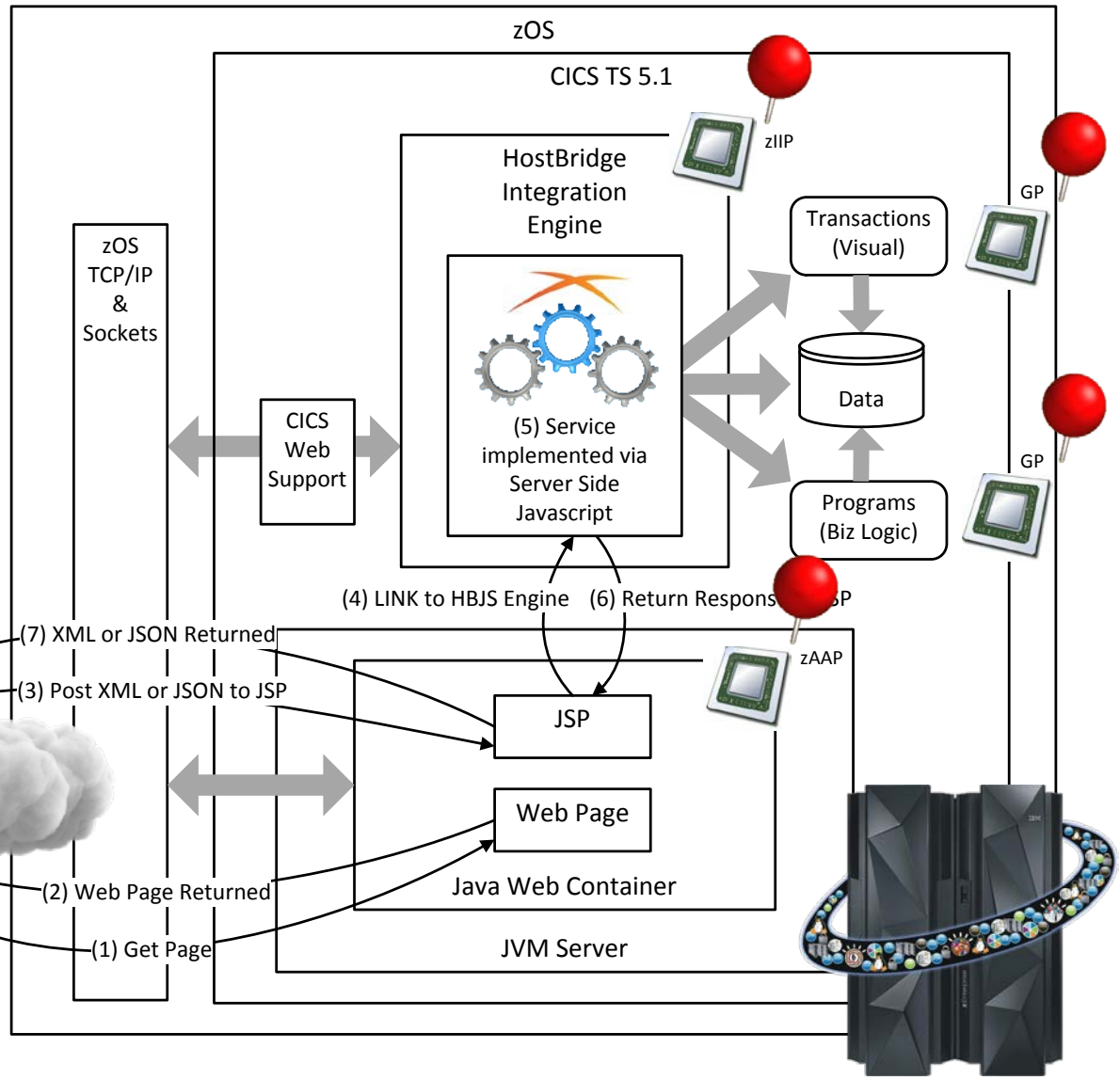
Java Web Container Config

Javascript exploited on both client AND server



Java Web Container Config

Maximum horizontal scalability by leveraging all System z capabilities



Operational Efficiency Case Study

- **Customer A**
  - **Industry: Telecommunications (US)**
  - **Very high daily/consistent transaction volume**
  - **Long-standing investment in COBOL-based socket apps**
- **Customer B**
  - **Industry: Financial Services (International)**
  - **Very high transaction volume on one day each month (and in compressed time period)**
  - **Long-standing investment in PL/I-based socket apps**

Common Objectives

- **Both customers had common objectives**
- **Business Objectives**
  - **Respond to competitive pressures in their industry**
  - **Lower incremental cost of high-volume CICS application processing (i.e., marginal value > marginal cost)**
  - **Move new/additional workload to System z and reinforce CICS TS as the most cost effective platform for their business**
- **Technical Objective (at least their hope)**
  - **Streamline System z and CICS integration paths**
  - **Reduce the CPU burn (GP) associated with socket applications and infrastructure**
  - **“Make the plumbing less expensive”**

Timing Was Opportune

- **Customers are continuing to state their concern about doing more for less (i.e., operational efficiency)**
- **We had just delivered zIIP-enabled versions of our products, and our heads were filled with fun facts related to:**
  - **z/OS, USS, LE, WLM, zIIP**
  - **CICS TS Open Transaction Environment**
  - **Sockets**

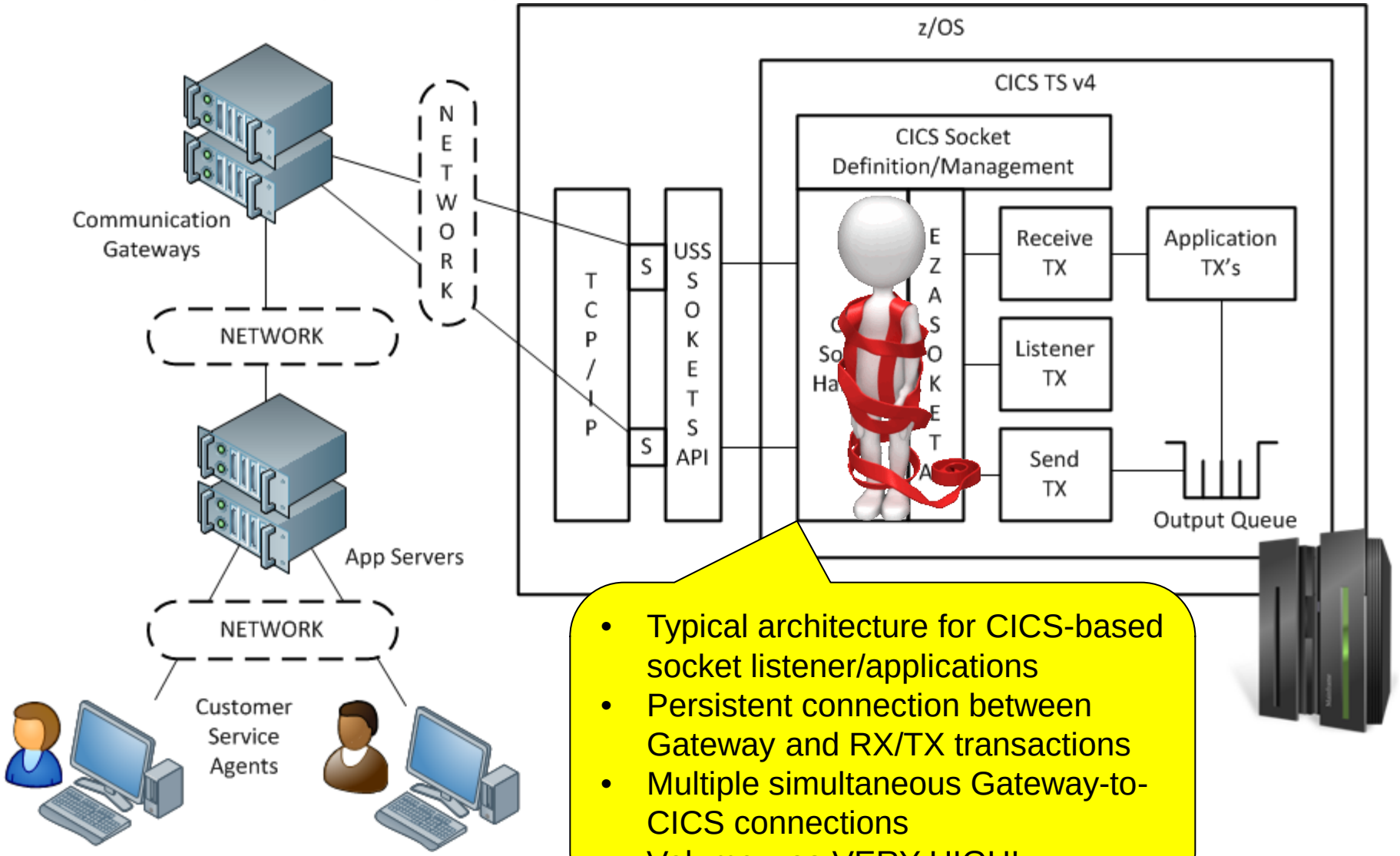
- **Other factors:**
  - **We are zealots regarding integration of CICS apps/data as part of web/cloud-based infrastructure**
  - **We are committed to delivering functionality under CICS**

Before the Deep Dive…

- **What we learned was surprising and the results were unexpected (in a good way)**
- **We ended up exploiting CICS TS OTE and z/OS to create a solution**
- **I want this to be knowledge you can use:**
  - **The approach is generally applicable to any CICS customer who has socket apps**
  - **The higher your volume, the more it matters**
- **Yes… I'm "a vendor" but please forget that for now – I'm speaking as an enthusiastic CICS developer**

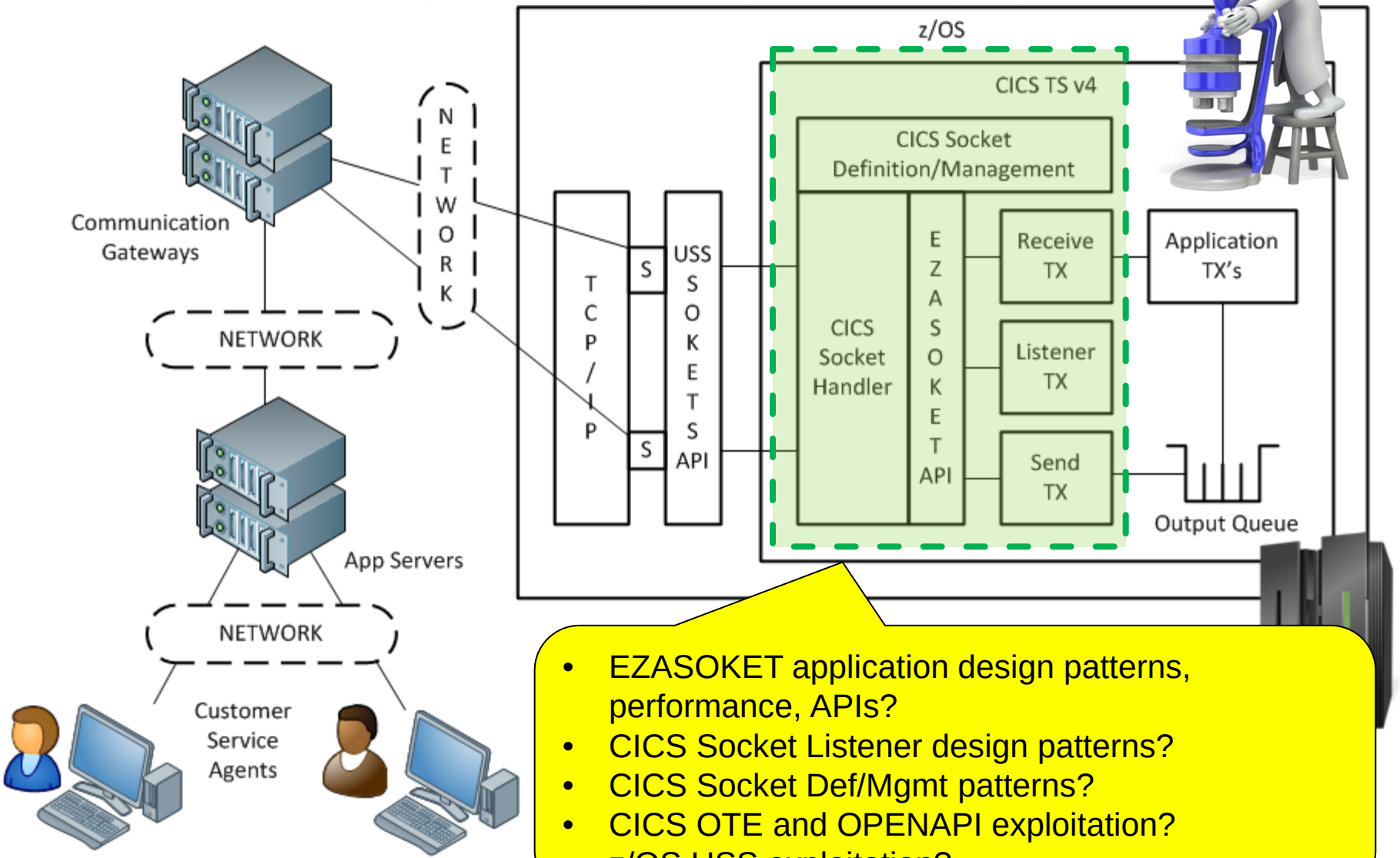
You can do this too.

Customer A - Initial Conditions



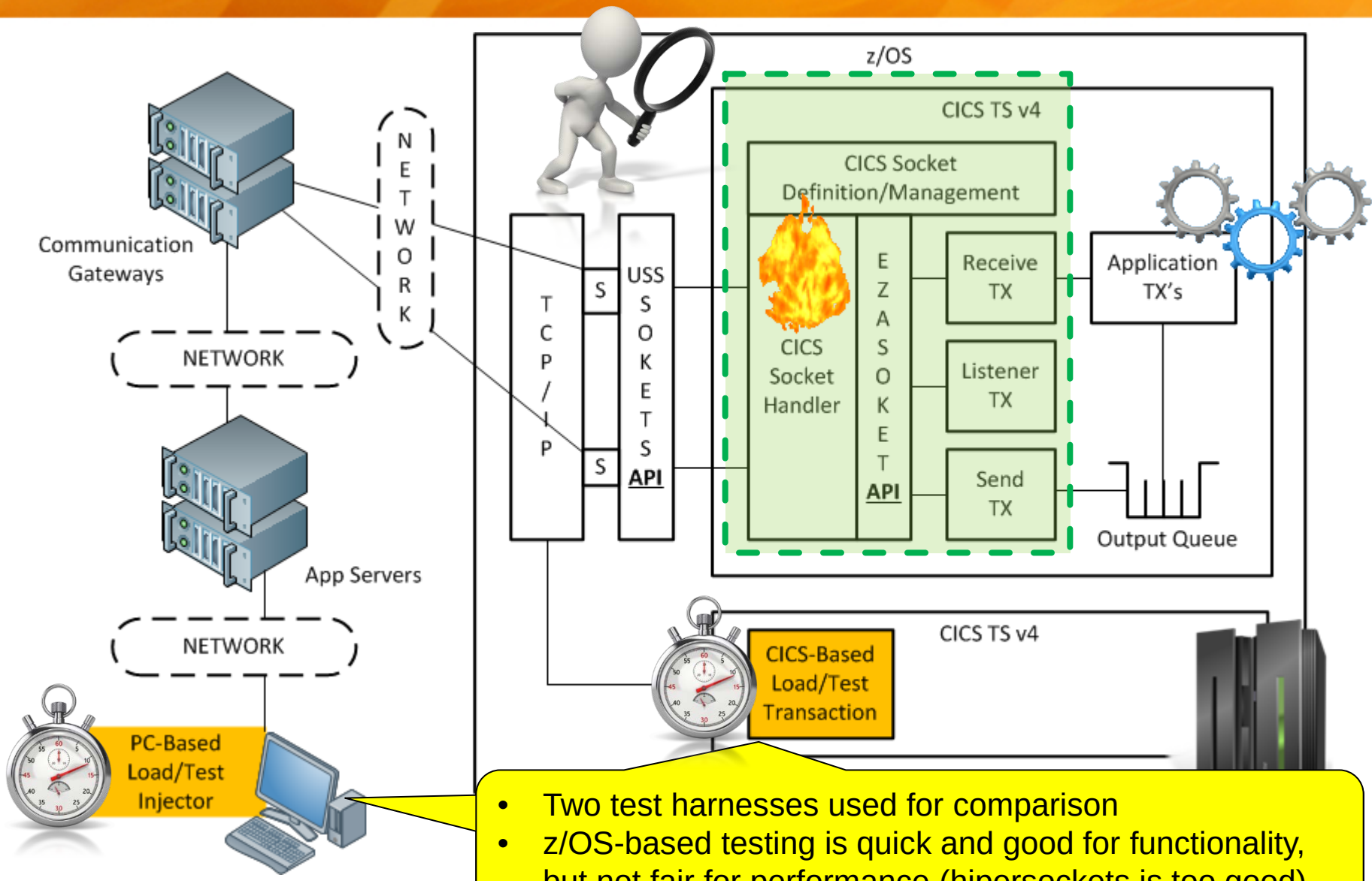
- Typical architecture for CICS-based socket listener/applications
- Persistent connection between Gateway and RX/TX transactions
- Multiple simultaneous Gateway-to-CICS connections
- Volume was VERY HIGH!

So We Examined This …



- EZASOKET application design patterns, performance, APIs?
- CICS Socket Listener design patterns?
- CICS Socket Def/Mgmt patterns?
- CICS OTE and OPENAPI exploitation?
- z/OS USS exploitation?

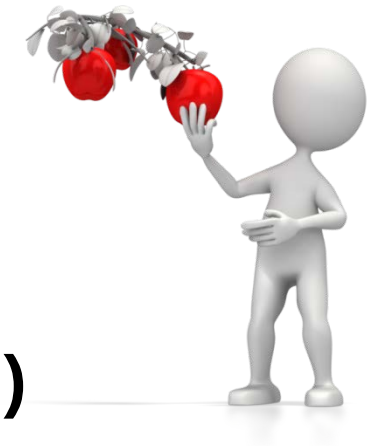
… by Running a Lot of Tests



- Two test harnesses used for comparison
- z/OS-based testing is quick and good for functionality, but not fair for performance (hipersockets is too good)

Where the Data Led Us

- **Under volume testing, the CPU burn associated with the CICS Sockets Support was measurable and linear (confirmed customer's theory)**
- **I won't characterize it as "high" or "low" because the only thing that mattered was whether it could be lower (or not so linear)**
- **Thus, we began to:**
  - **Isolate various components and their impact**
  - **Consider how to provide alternative functionality (but complimentary to CICS TS)**
- **Low hanging fruit seemed to be CICS Socket Handler (via EZASOKET API)**

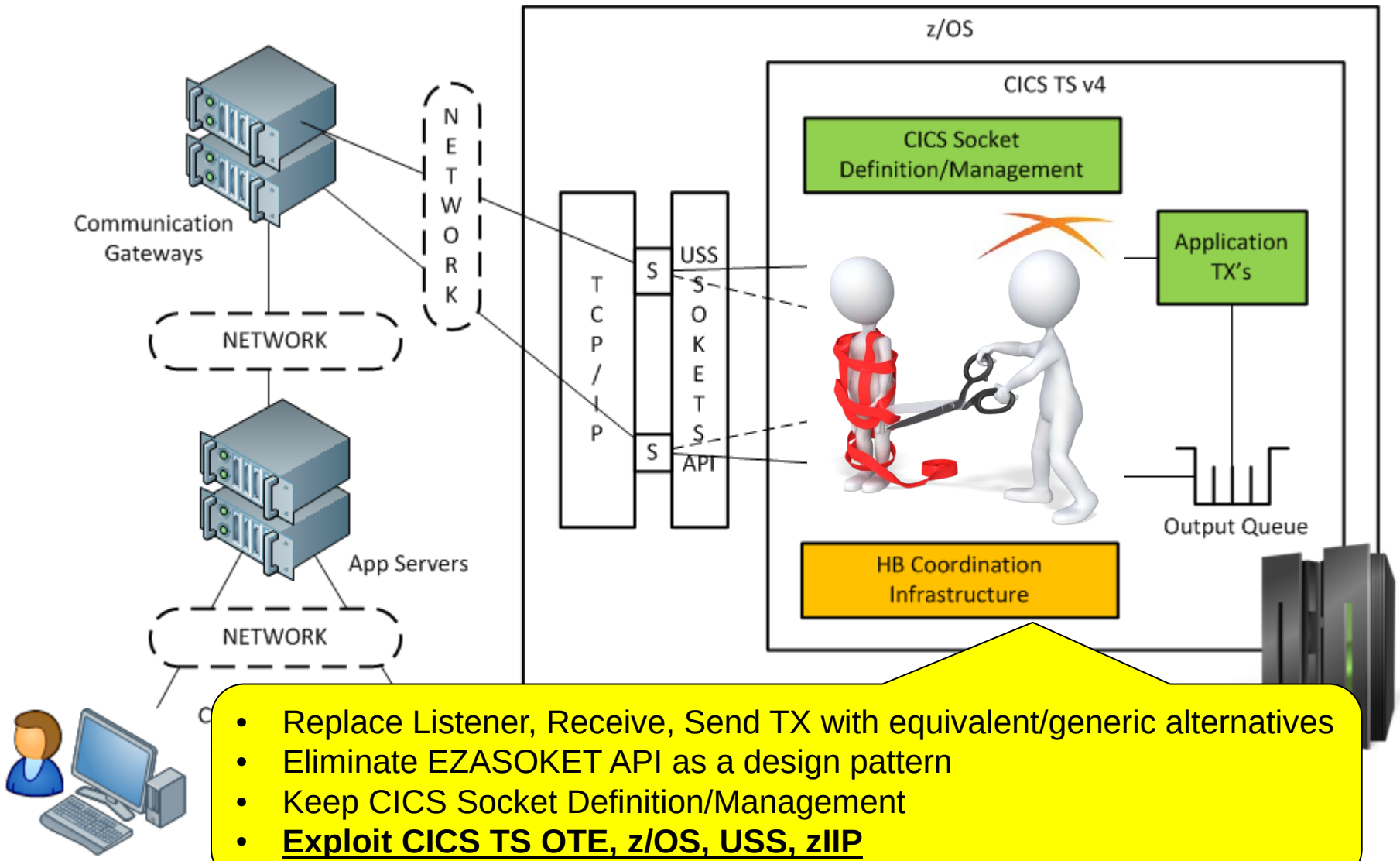


“CICS Socket Support”

- **Provided as part of z/OS Communications Server**
- **What it includes:**
  - **Socket APIs (aka, EZASOKET or EZACICSO)**
  - **Listeners: standard and enhanced (i.e., CSKL)**
  - **Definition and management components (e.g., EZAO)**
- **A well-documented workhorse, but…**
- **It’s been around a long time (circa 1992)**
- **Older than CICS OTE**
  - **Thus… much of it’s original architecture**
- **Reengineered to support OTE**
  - **But… the general approach of the original architecture persisted**
- **However… much has changed in zOS and CICS TS!**

Thus, I’m NOT referring to CICS TS features which use the CICS Sockets Domain.

Customer A – Solution



- Replace Listener, Receive, Send TX with equivalent/generic alternatives
- Eliminate EZASOKET API as a design pattern
- Keep CICS Socket Definition/Management
- **Exploit CICS TS OTE, z/OS, USS, zIIP**

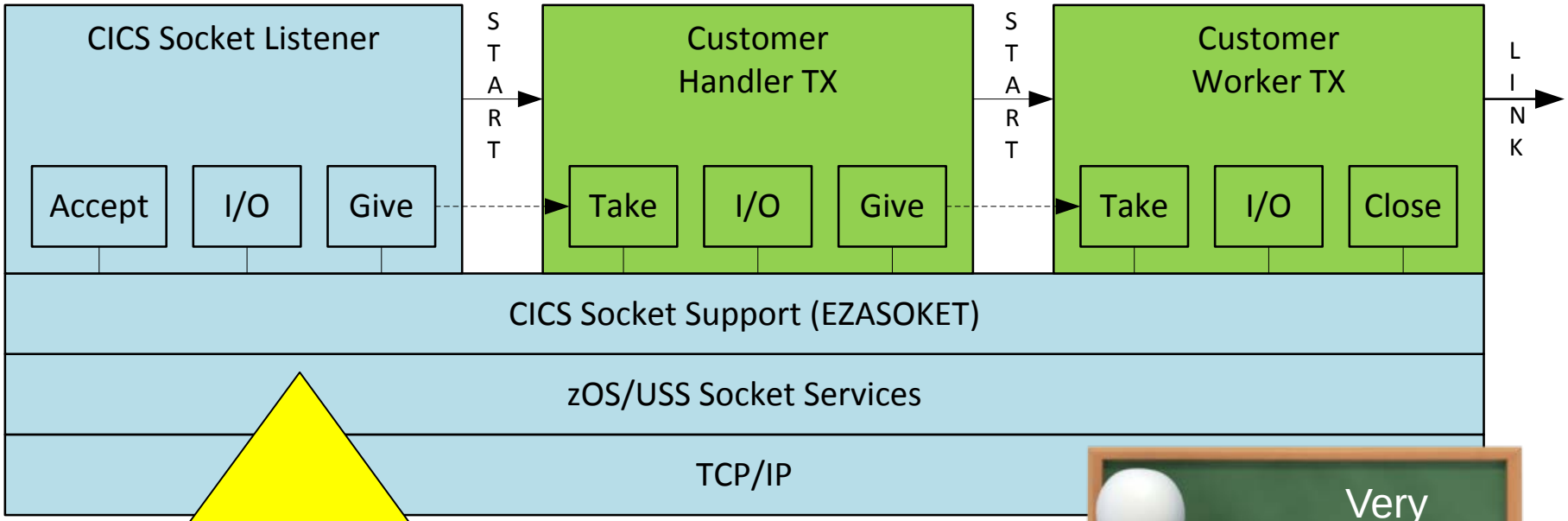
Solution Assessment

- **Excellent…**
  - **GP CPU burn associated with Socket I/O went way down (40-45%)**
  - **EZASOKET API eliminated (all components use native sockets)**
  - **Transparent to the customer's applications**
  - **CICS Socket definition/management leveraged**
    - **EZAO still used to Configure, Start, or Stop Listeners**

- **zIIP enablement potential maximized**
  - **HostBridge Socket Support code is zIIP enabled**
  - **Customer application code not zIIP enabled (per IBM-ISV T&C's)**
  - **Minimal task switching**

Customer B - Initial Conditions

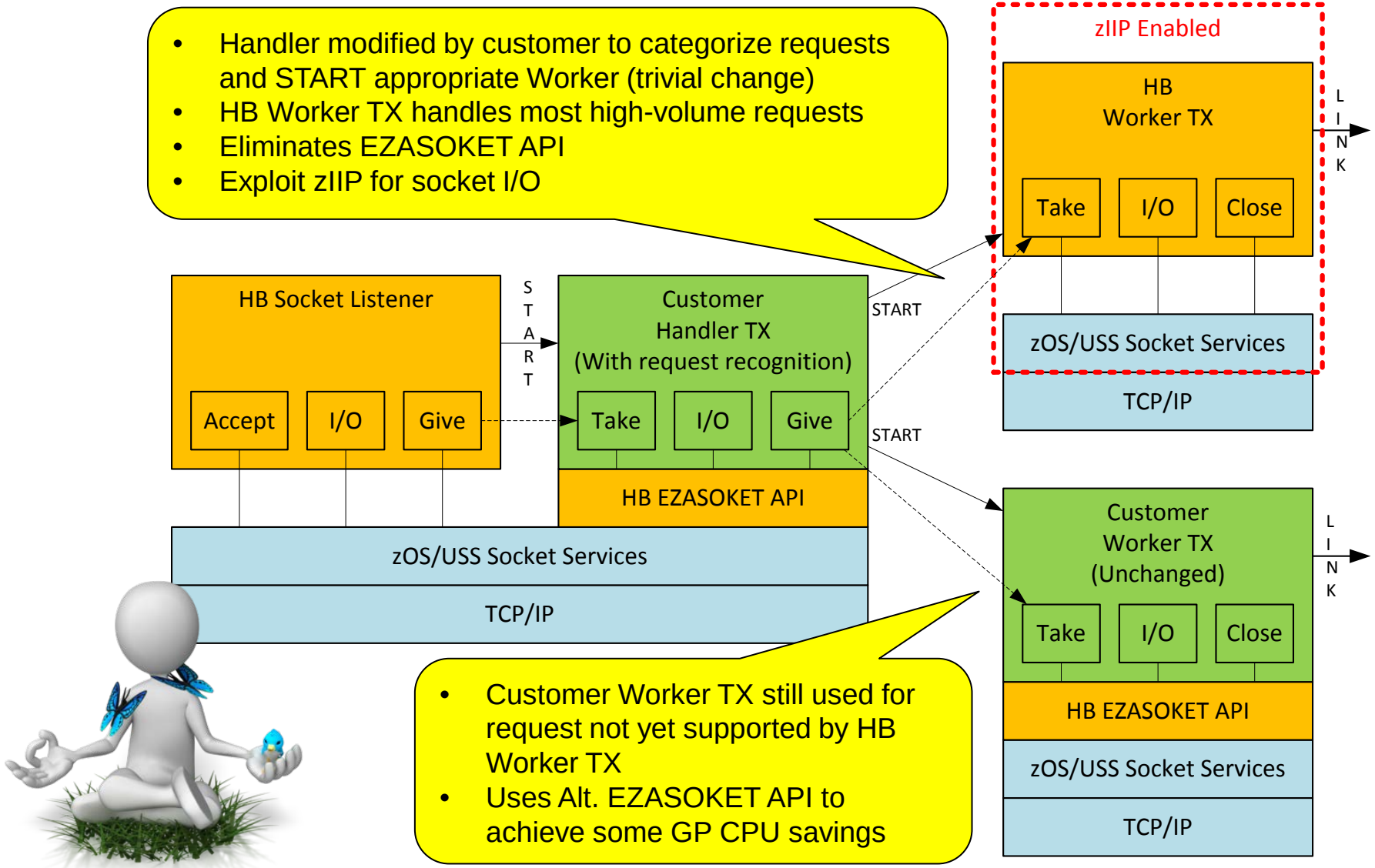
(Infrastructure outside System z similar to customer A)



- A single socket is used for both sending and receiving
- CICS Socket Listener connection requests
- Handler TX validates and categorizes requests
- Worker TX is long-lived
- Work requests serviced by LINKed-to programs

Customer B – Solution

- Handler modified by customer to categorize requests and START appropriate Worker (trivial change)
- HB Worker TX handles most high-volume requests
- Eliminates EZASOKET API
- Exploit zIIP for socket I/O



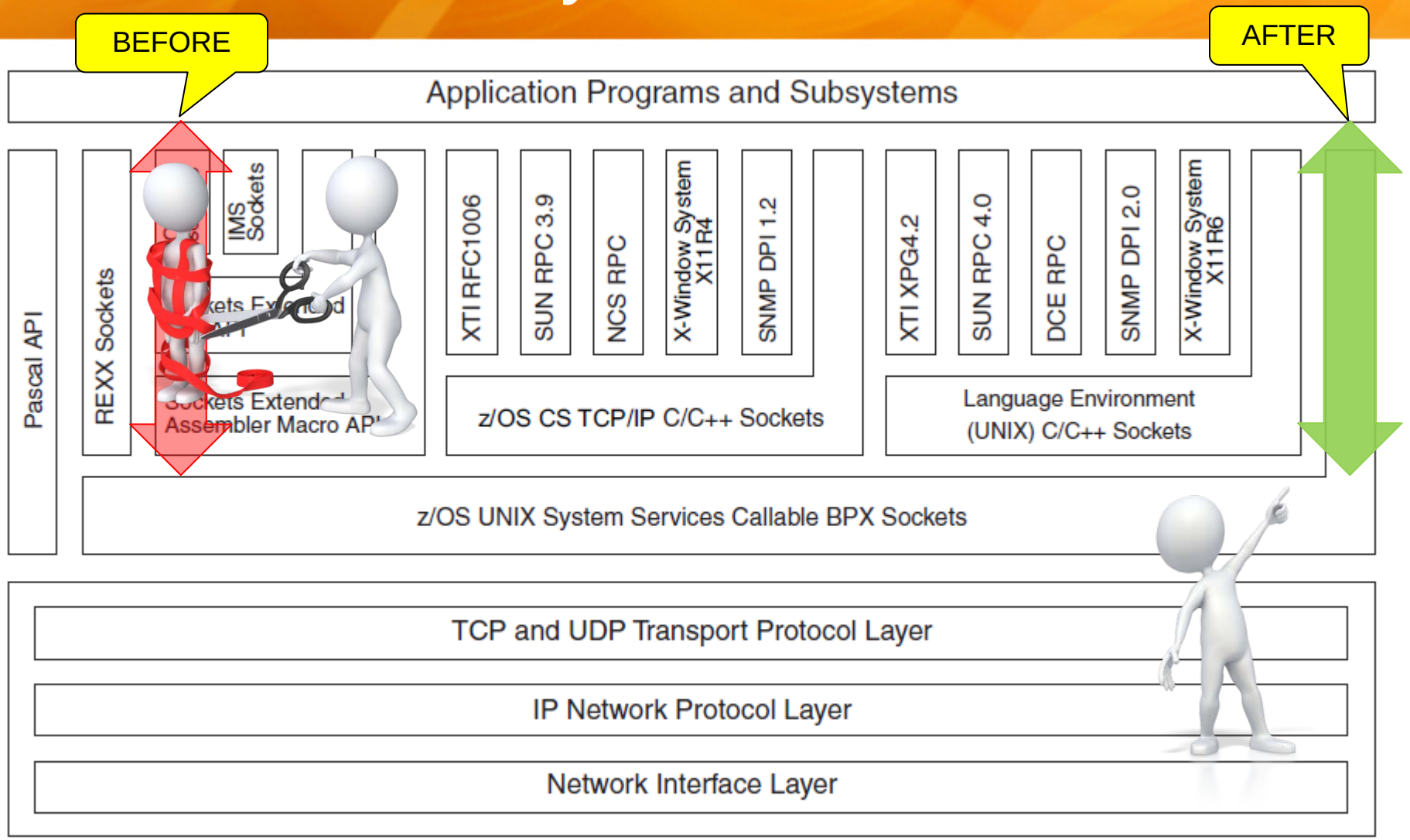
- Customer Worker TX still used for request not yet supported by HB Worker TX
- Uses Alt. EZASOKET API to achieve some GP CPU savings

Value Proposition Model

- **What mattered most to the customer was processing new workload efficiently during their peak 4 hour period**
- **Assume:**
  - **5 million TX in max 4 hour period**
  - **20% processed via HB Worker TX**

5,000,000	Peak 4 hour transaction volume
20%	% of TX processed via HB Worker
1,000,000	TX processed via HB Worker
80%	% of TX processed via Std. Worker
4,000,000	TX processed via Std. Worker
903	Est. GP CPU Reduction for HB Worker (seconds)
807	Est. GP CPU Reduction for Std. Worker (seconds)
1,710	Total Est. GP CPU Seconds Reduced
28.49	Total Est. GP CPU Minutes Reduced during Peak Period

Pathway - Old vs. New



z/OS Communications Server, IP Sockets Application Programming Interface Guide and Reference

Tooling Developed

- **It's difficult to get a snapshot of a CICS region's total resource consumption that is:**
  - **high-resolution (microseconds)**
  - **low-overhead**
  - **Immediate**
  - **Includes zIIP and zAAP**
- **Ended up developing two tools:**
  - **A CICS transaction to provide a summary of MVS ASSB timers (HBZT)**
  - **A CICS XMNOUT exit to log transaction metrics via WTO**
- **The combination allowed us to:**
  - **drive testing fast**
  - **quickly assess results from all angles**
- **Special thanks to:**
  - **Larry Lawler (UNICOM)**
  - **Ed Jaffe (Phoenix Software)**
- **For info on HBZT, see me after session (it's free)**

CPU Measurement (HBZT)

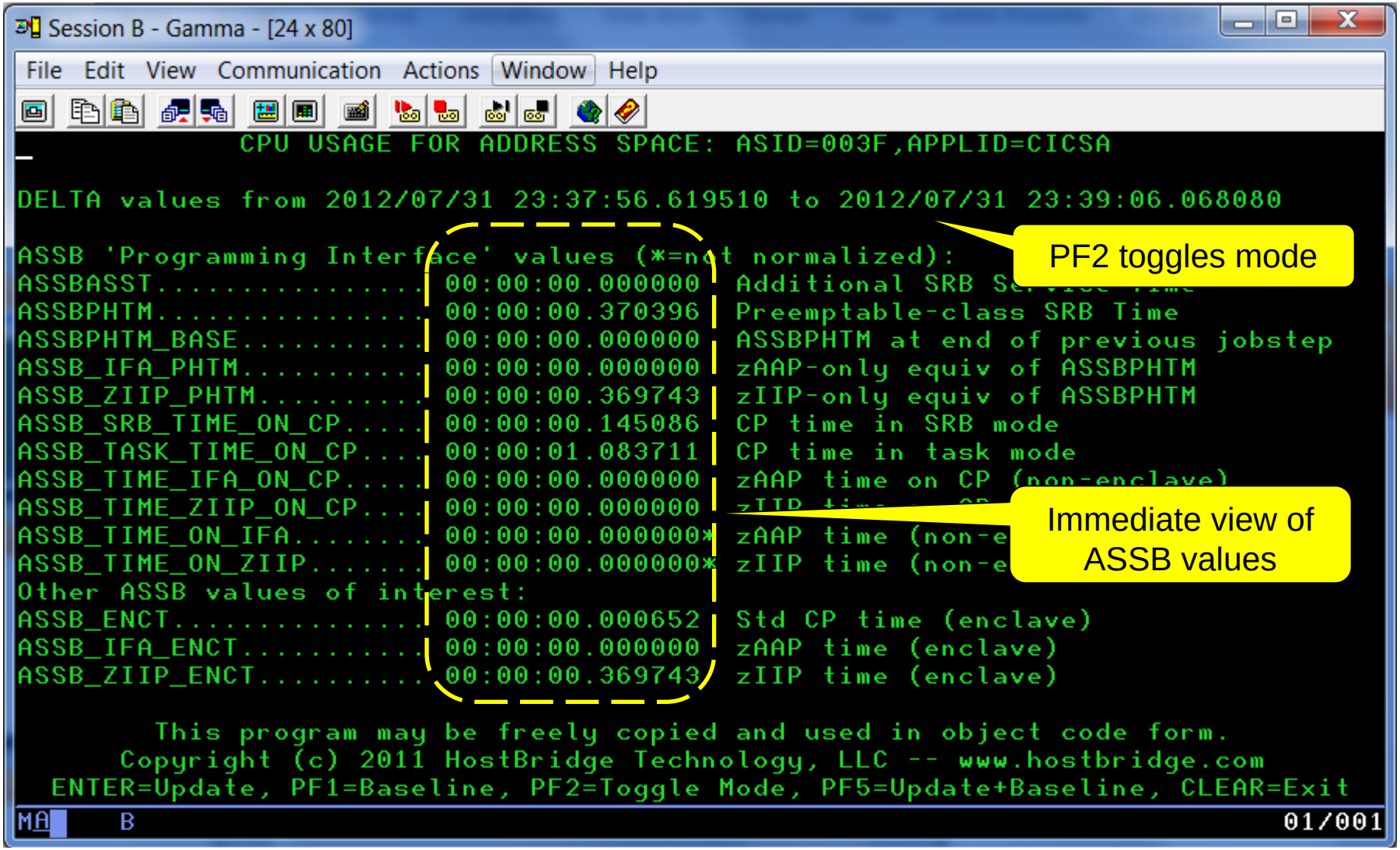
Session B - Gamma - [24 x 80]

File Edit View Communication Actions Window Help

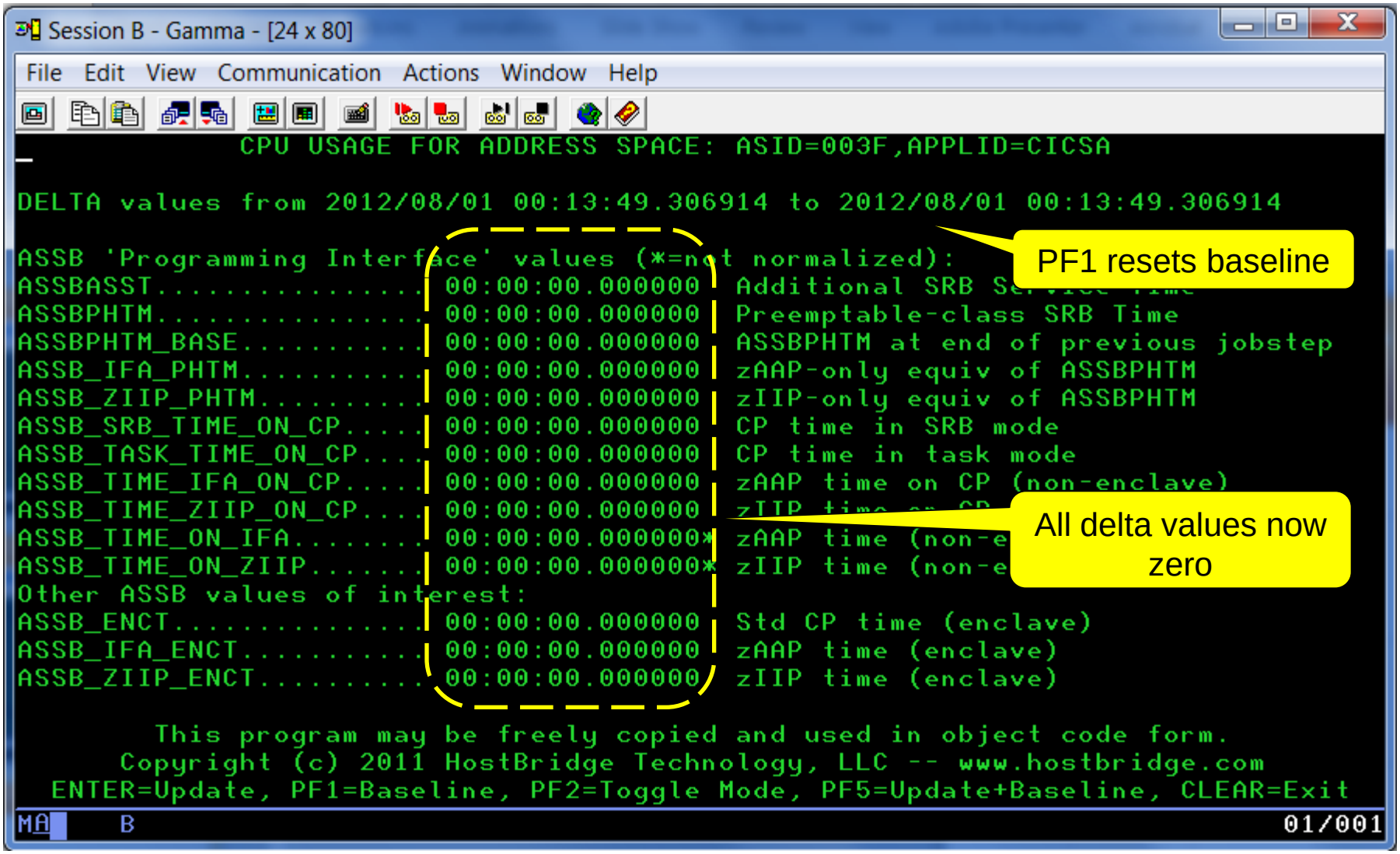
```
            CPU USAGE FOR ADDRESS SPACE: ASID=003F,APPLID=CICSA

ACTUAL values at 2012/07/31 23:39:06.068080

ASSB 'Programming Interface' values (*=not normalized):
ASSBASST.................  00:00:00.000000  Additional SRB Service Time
ASSBPHTM.................  00:00:00.385582  Preemptable-class SRB Time
ASSBPHTM_BASE............  00:00:00.000000  ASSBPHTM at end of previous jobstep
ASSB_IFA_PHTM............  00:00:00.000000  zAAP-only equiv of ASSBPHTM
ASSB_ZIIP_PHTM...........  00:00:00.378829  zIIP-only equiv of ASSBPHTM
ASSB_SRB_TIME_ON_CP......  00:00:00.288598  CP time in SRB mode
ASSB_TASK_TIME_ON_CP.....  00:00:02.473032  CP time in task mode
ASSB_TIME_IFA_ON_CP......  00:00:00.000000  zAAP time on CP (non-enclave)
ASSB_TIME_ZIIP_ON_CP.....  00:00:00.000000  zIIP time on CP (non-enclave)
ASSB_TIME_ON_IFA.........  00:00:00.000000* zAAP time (non-enclave)
ASSB_TIME_ON_ZIIP........  00:00:00.000000* zIIP time (non-enclave)
Other ASSB values of interest:
ASSB_ENCT................  00:00:00.006224  Std CP time (enclave)
ASSB_IFA_ENCT............  00:00:00.000000  zAAP time (enclave)
ASSB_ZIIP_ENCT...........  00:00:00.378829  zIIP time (enclave)

        This program may be freely copied and used in object code form.
      Copyright (c) 2011 HostBridge Technology, LLC -- www.hostbridge.com
  ENTER=Update, PF1=Baseline, PF2=Toggle Mode, PF5=Update+Baseline, CLEAR=Exit
MA    B                                                          01/001
```

ACTUAL mode upon entry

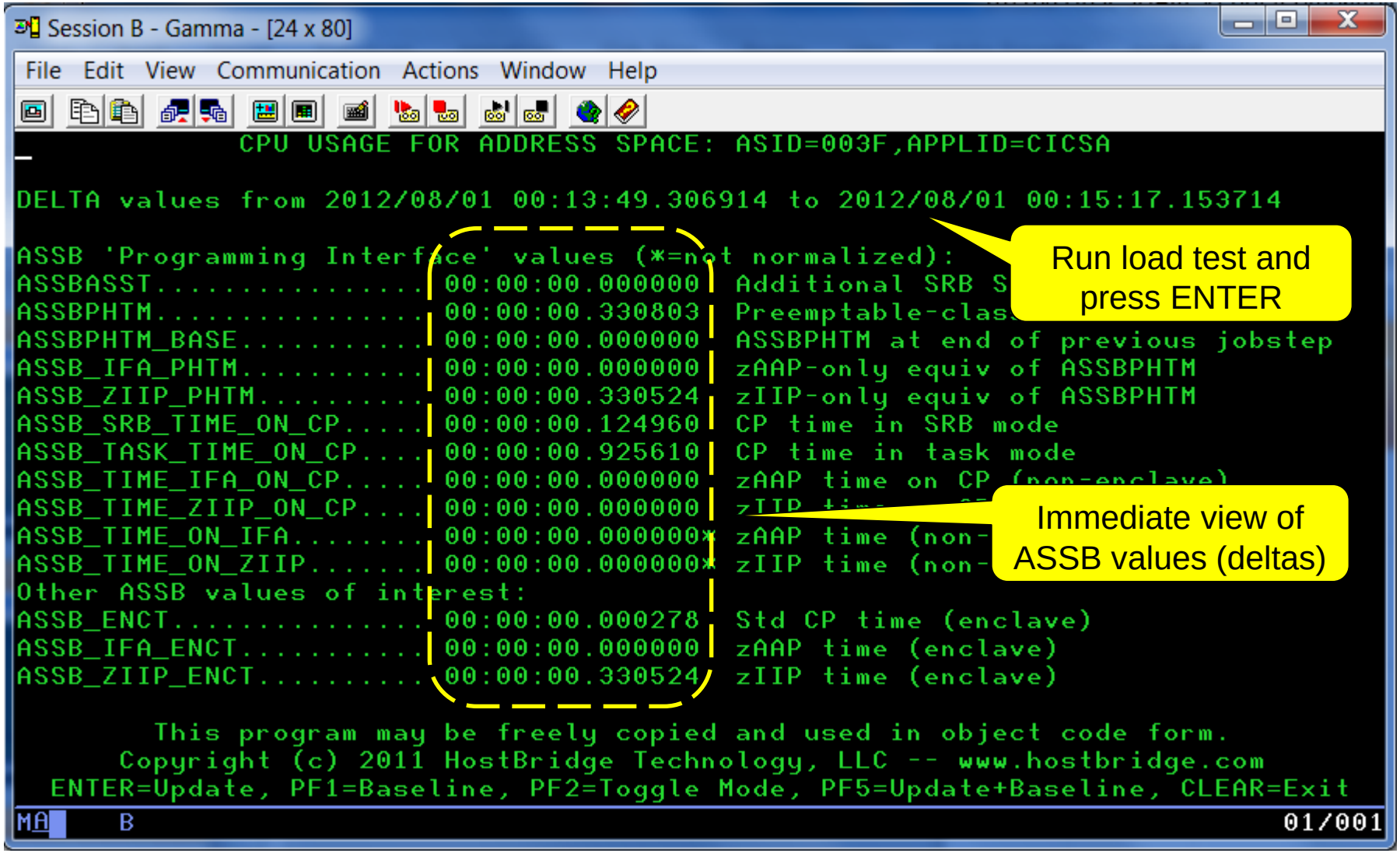
CPU Measurement (HBZT)



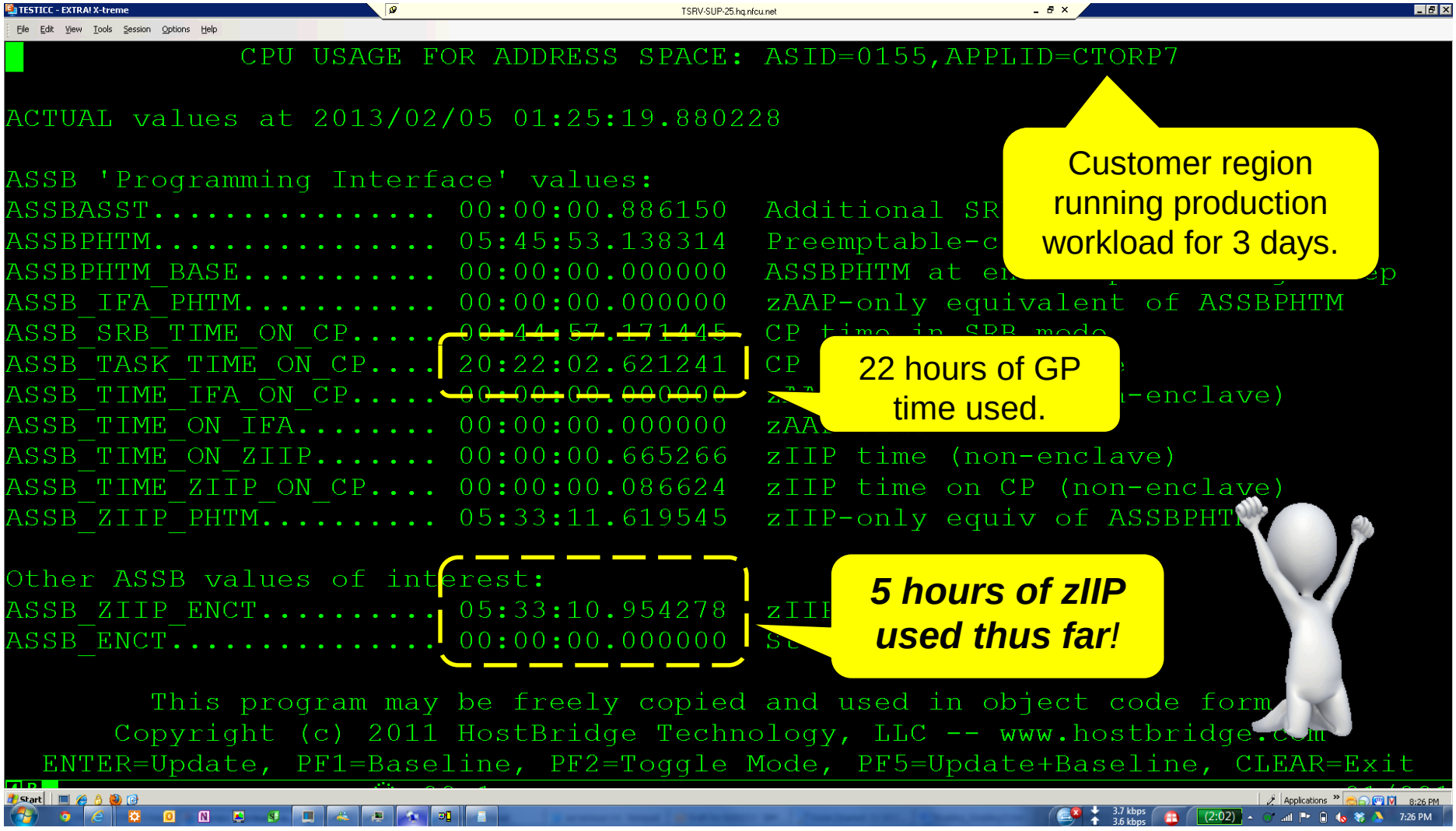
CPU Measurement



CPU Measurement



CPU Measurement



Case Studies - Summary

- **Operational efficiency is paramount to all System z customers**
- **The CICS TS Open Transaction Environment continues to evolve and creates new opportunities for customers and ISV's to extract savings**
- **The approach embodied by HostBridge Socket Support is just an example of what's possible**
  - **Applicable to any customer who uses CICS Socket Support**
  - **zIIP support can only be provided by a licensed ISV**
- **Bottom Line: There is no reason to devote high-value MIPS to integration or "plumbing"**
- **Oh… and the customers were very pleased**

You can do this too… and more.

Wrap Up

- **CICS TS 5.1 is a ground-breaking release**
  - **HostBridge 6.62 is our corresponding support release**
- **From the perspective of an integration-minded ISV, what excites us the most are:**
  - **Continued OTE enhancements**
  - **Ongoing 64-bit Storage enhancements**
  - **Significant Java performance enhancements**
  - **The WAS Liberty Profile (i.e., the Web Container)**
- **Poor integration solutions have often given System z apps a bad rap**
- **Customers who take a fresh look at modern CICS and System z capabilities will be rewarded!**

WANTED: Tales from the Trenches

- **We KNOW you are doing cool stuff (we see it every week)**
- **We are looking for Tales from the Trenches to SHARE next time**
- **Practical stories about how you are:**
  - **Meeting business challenges**
  - **Overcoming technical hurdles**
  - **Transforming your CICS apps for the future**
  - **Leveraging new features of CICS**
- **You can tell your Tale yourself, or someone will do it for you (it can be completely anonymous)**
- **The objective is to create an active feedback loop of user experiences within the CICS community**