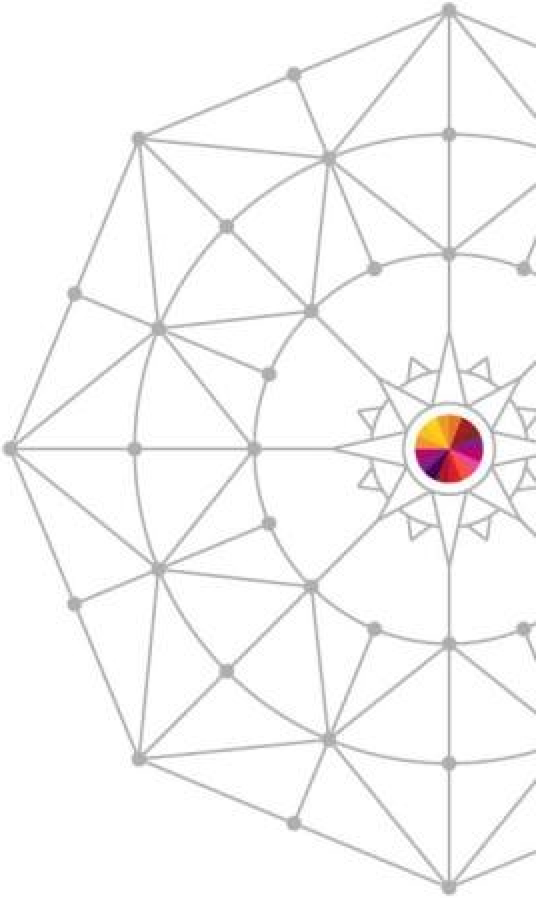
TPFUG
Washington, D.C.

z/TPF HTTP Support

Mark Gambino
TPF Development Lab

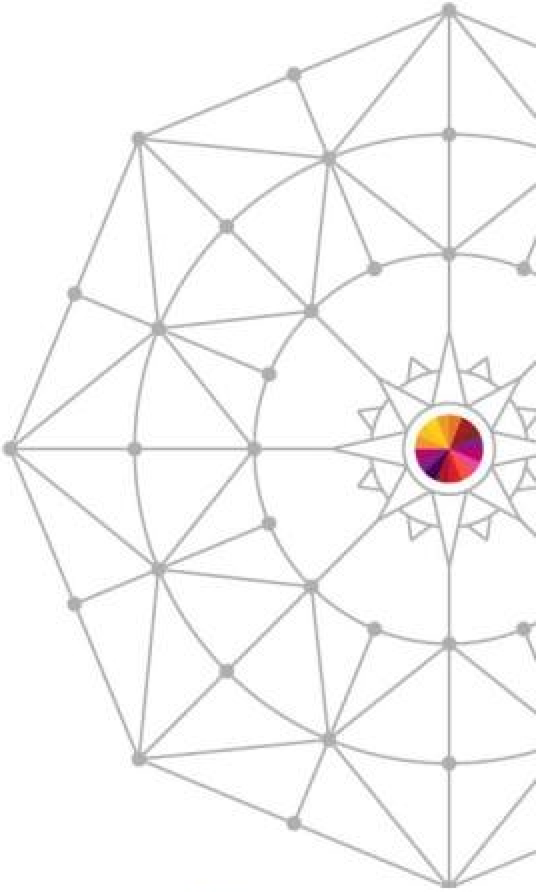
Communications Subcommittee
March 10, 2014



dW ibm.biz/zTPF_dW | @IBM_TPF #IBMTPF #TPFUG

Disclaimer

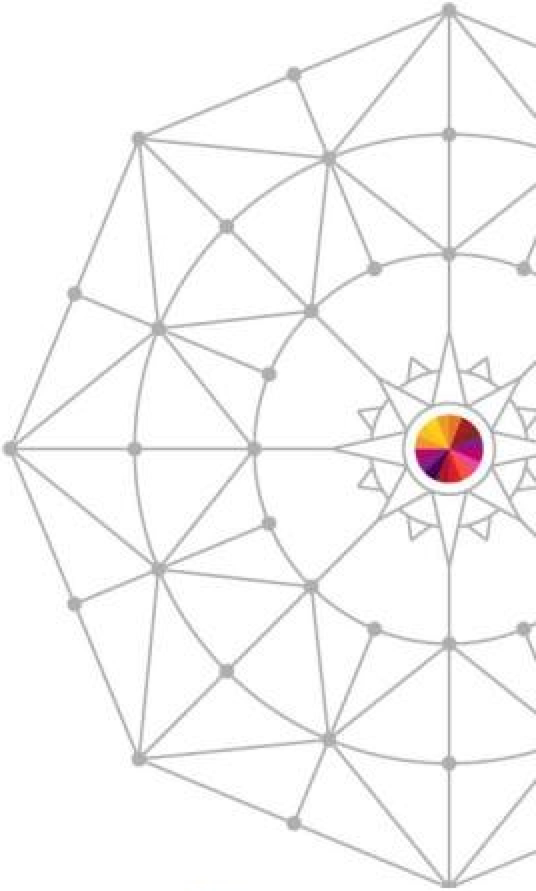
Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.



TPFUG
Washington, D.C.

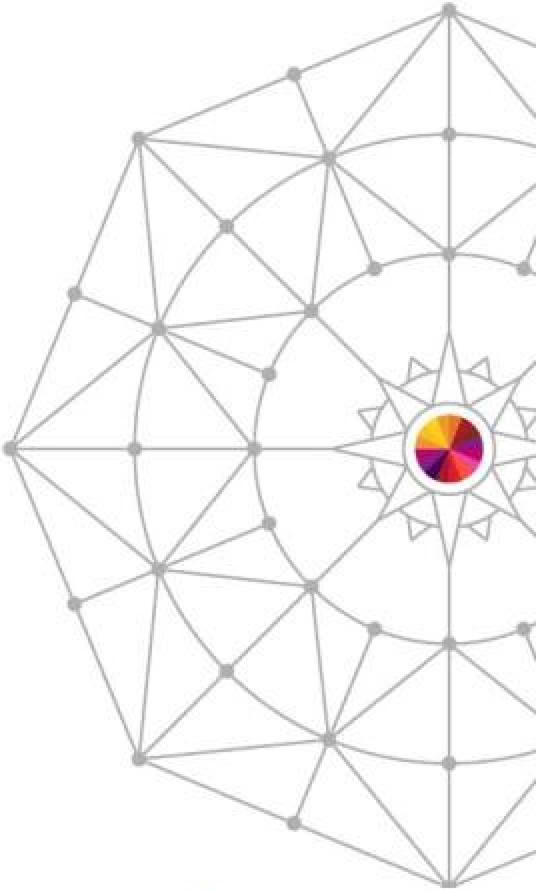
HTTP Client Support on z/TPF

- APARs PJ32013 and PJ32052 (PUT 4)
- Provided APIs to enable z/TPF applications to send HTTP requests to remote HTTP servers
- Processing was synchronous
  - Application ECB that issues the API is suspended until the response is received from the HTTP server
- Uses the HTTP client library within libcurl
- Can send requests over regular HTTP or secure HTTP (HTTPS – HTTP over SSL)



Advanced HTTP Client Support on z/TPF

- APARs PJ34208, PJ37296 PJ37454 (PUT 7)
  - Allowed for asynchronous programming model where ECB that issues the HTTP client API can exit and a new ECB is created when the HTTP response is received
- APARs PJ41866 and PJ41867
  - Performance enhancements to reduce amount of CPU used by HTTP client support
  - Improved scalability
    - Increased number of shared sessions supported
    - Maximum request queue limits expanded
  - Fixes to various problems also included



TPFUG Washington, D.C.

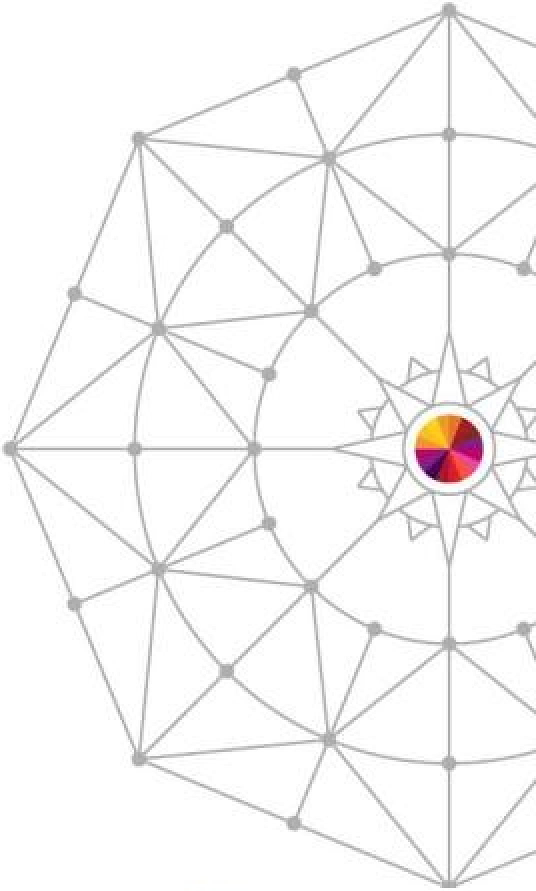
HTTP Client POST/PUT Flows

- The default behavior of libcurl of an HTTP POST or PUT request requires 4 flows:
  1. Client sends HTTP header with “Expect 100” in the HTTP header along with the request URL
  2. Server responds with HTTP status code 100 (continue)
  3. Client sends the HTTP request message
  4. Server responds with HTTP status code 200 and the HTTP response message
- The first message exchange is done to verify that the target exists before sending the request message

Optimized HTTP Client POST/PUT Flows

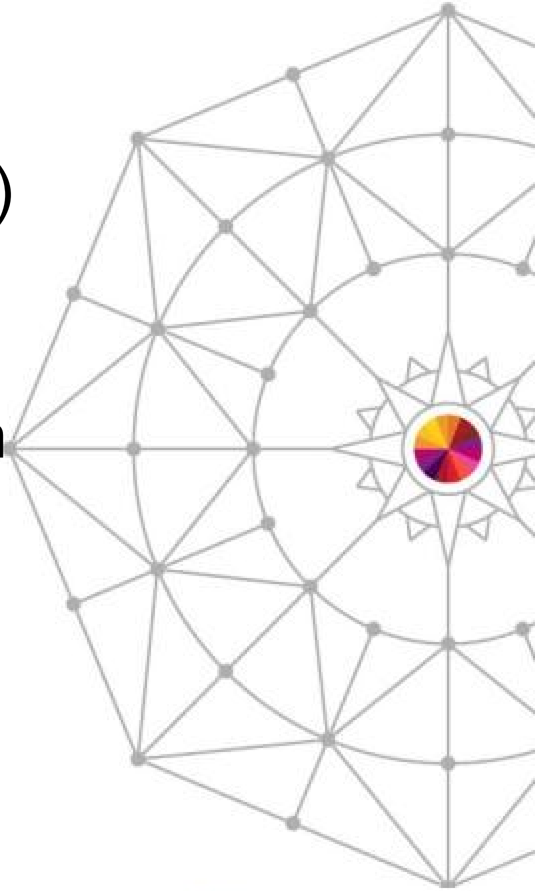
- Application program can override the default behavior to reduce the number of message flows
- Application program issues the *tpf_httpPerform* API that includes an empty expect header “Expect:”
- This results in 2 flows:

1. Client sends the entire HTTP request
2. Server responds with HTTP status code 200 and the HTTP response message



Apache HTTP Server

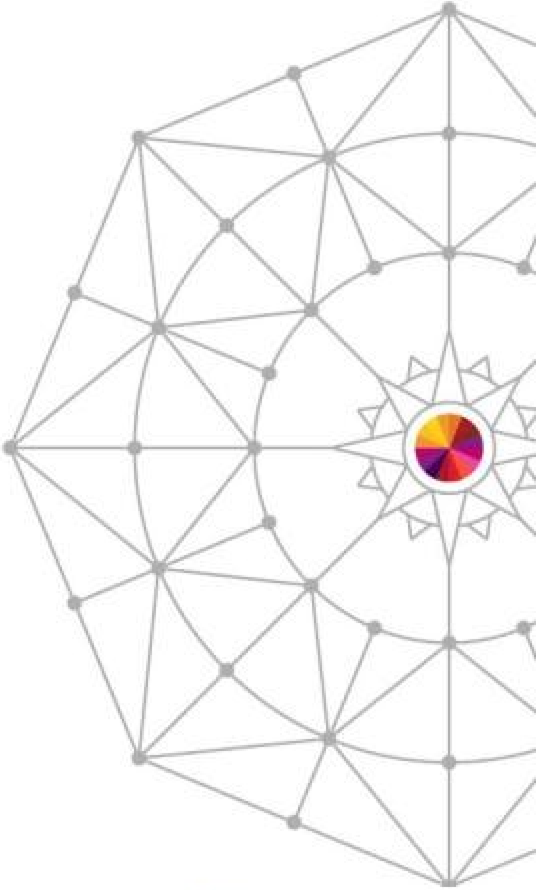
- TPF 4.1 supported the Apache HTTP Server 1.3
- z/TPF APARs PJ34337 and PJ34514 (PUT 5) added support for the Apache HTTP Server 2.2
- Apache is a very robust HTTP server that can be customized with many optional plug-in modules to perform various functions
  - For example, if you build Apache with the mod_ssl module enabled, this allows you to use secure HTTP server (HTTP over SSL)



TPFUG Washington, D.C.

z/TPF HTTP Server Support

- APARs PJ39252 and PJ39550 (PUT 9)
- Came about because of TPFUG requirement "SOA00002 – *Lightweight HTTP Server for SOAP Messaging"*
- Subset of the HTTP to be an efficient message transport
- Allows for asynchronous programming model where one ECB receives an HTTP request and a different ECB can send the HTTP response



TPFUG Washington, D.C.

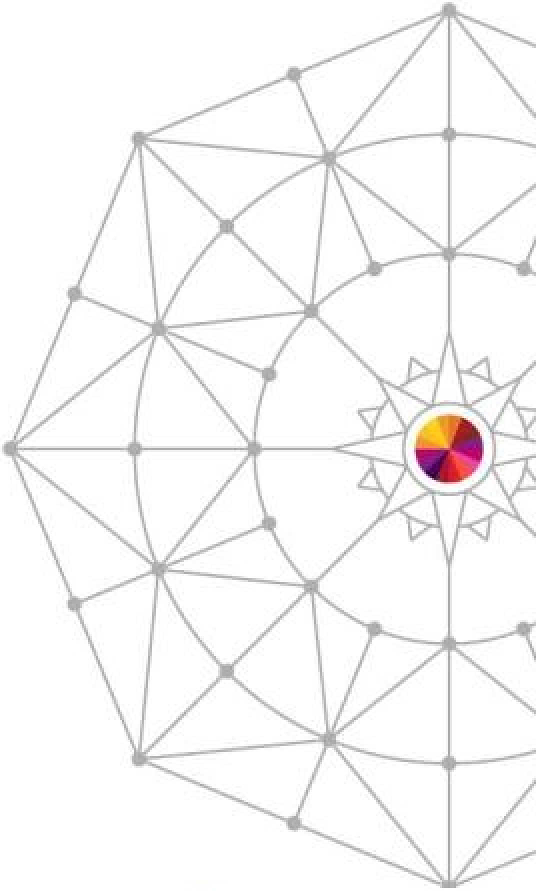
z/TPF Secure HTTP Server Support

- APARs PJ41171 (PUT 10) and PJ41711
  - PJ42080 fixes a base problem you might encounter using secure HTTP server
- Adds HTTPS (HTTP over SSL) support to the z/TPF HTTP server
- Define to z/TPF using the new INETD SSL model (APAR PJ41170)
  - Specify PGM-CHS1 on ZINET ADD command to define a secure HTTP server application
- Added network services database (NSD) support for all z/TPF HTTP servers (HTTP and HTTPS)
- Interface between z/TPF HTTP server layer and application server program is the same for both HTTP and HTTPS servers
  - Easy to migrate from HTTP to HTTPS

TPFUG Washington, D.C.

Regular vs Secure HTTP Server Performance Comparison

- HTTP client driver was run using different message sizes over persistent HTTP sessions
  - – First test sent regular HTTP traffic
  - – Second test sent secure HTTP traffic
- Server application on z/TPF used async model
  - – When ECB #1 received an HTTP request message it created ECB #2 that then sent the HTTP response
- z/TPF HTTP Server was running on an LPAR with 4 I-streams on z196, dedicated PR/SM



Regular vs Secure HTTP Server Performance Results

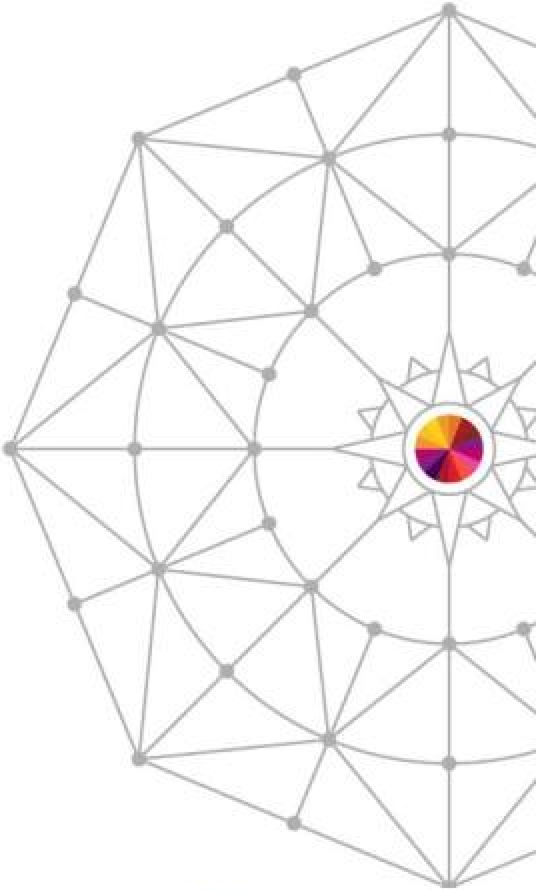
	Regular HTTP Traffic			Secure HTTP Traffic		
Message Size	Messages/ second	CPU Util	Message Rate at 100% CPU	Messages/ second	CPU Util	Message Rate at 100% CPU
100	49049	41.9%	117062	28892	46.2%	62537
1000	19557	72.2%	27087	12280	54.5%	22532
3000	9534	76.1%	12528	5097	56.3%	9053
5000	6705	80.9%	8288	3314	59.0%	5617

* Several factors influence performance. Your results may vary.

TPFUG Washington, D.C.

New INETD Model for SSL

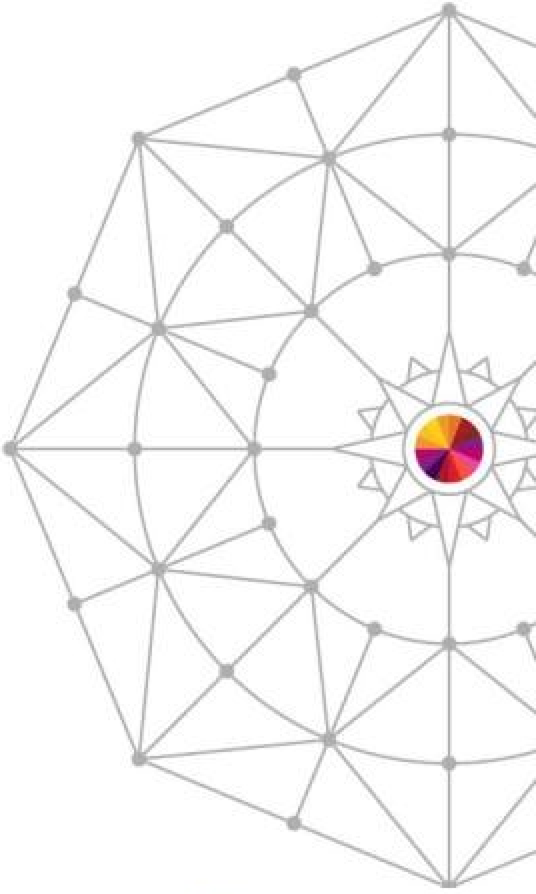
- APAR PJ41170 (PUT 10)
- New MODEL-SSL option for INETD servers
  - Similar to MODEL-AOA2, but adds SSL
- INETD manages all the TCP socket and SSL session establishment
- Application programs and middleware just need to send and receive data
  - Same as what application programs do for INETD TCP MODEL-NOWAIT servers for non-SSL sessions
  - Easy to convert a TCP MODEL-NOWAIT server to MODEL-SSL server



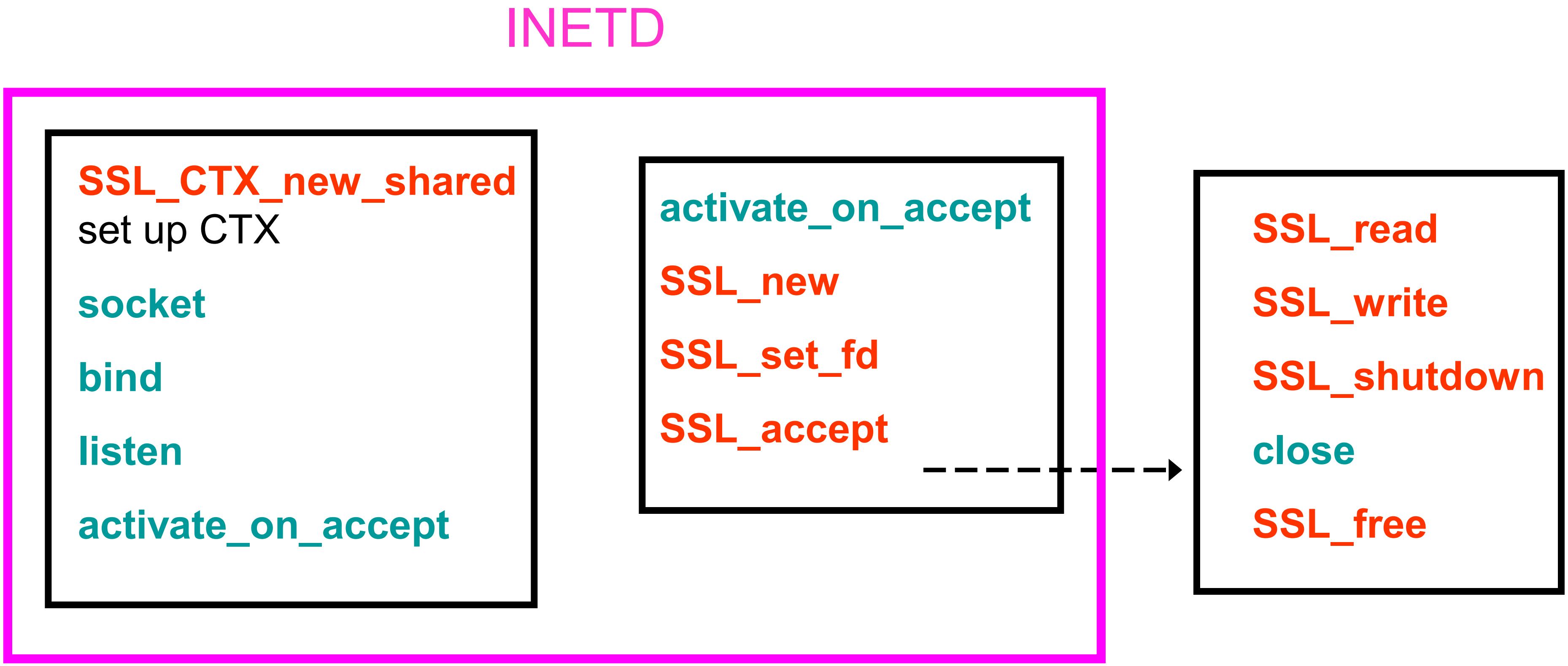
TPFUG Washington, D.C.

New INETD Model for SSL

- INETD responsibilities:
  - Create and manage the listener socket
  - Create a single CTX that can be used by all SSL sessions using this server application
  - Create SSL structures
  - Start SSL sessions
  - Automatically restart the listener socket if it fails
  - Automatically create a new CTX if the shared SSL daemons recycle
- Server application responsibilities:
  - Send and receive data over the SSL session
  - Shutdown the SSL session and close the connected socket



New INETD Model for SSL



INETD

SSL_CTX_new_shared
set up CTX

socket

bind

listen

activate_on_accept

activate_on_accept

SSL_new

SSL_set_fd

SSL_accept

SSL_read

SSL_write

SSL_shutdown

close

SSL_free

TPFUG
Washington, D.C.

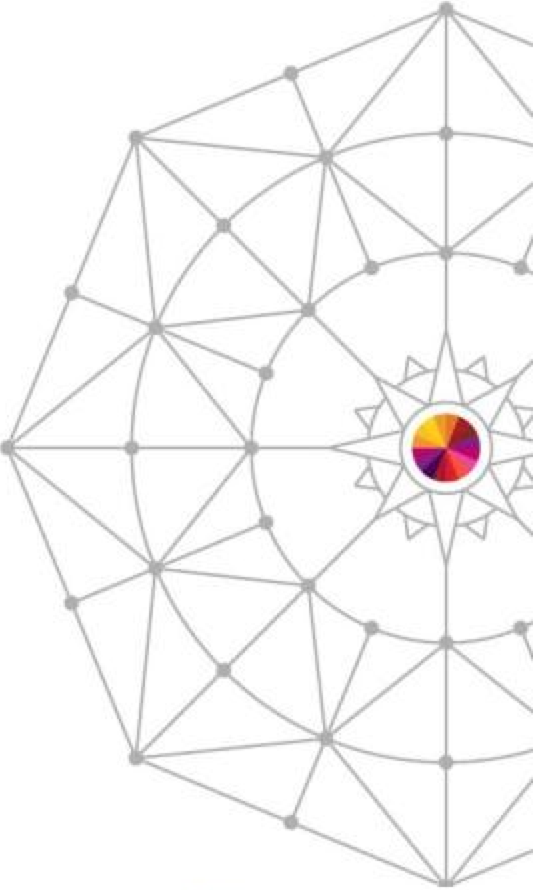
SSL Configuration Information New INETD Model for SSL

- SSL configuration information for the server application are defined using the existing *Application Configuration Files for SSL* mechanism
- Sample configuration file for SSL:

```
USESSL=YES
CIPHER=DES-CBC3-SHA
VERIFYPEER=NO
CERTIFICATE=/certs/tpfprodcert.pem
CERTTYPE=PEM
KEY=/tpfpubk/keypair1.pem
KEYTYPE=PEM
VERSION=TLSV1
```

New User Exit for New INETD Model for SSL

- INETD calls new user exit ussl.c when a MODEL-SSL application is starting or stopping
- ZINET START:
  - User exit allows you to initialize tables that are used by this application
  - User exit will be called before any SSL sessions are started
- ZINET STOP:
  - User exit will allow you to clean up tables that are used by this application



TPFUG Washington, D.C.

Questions

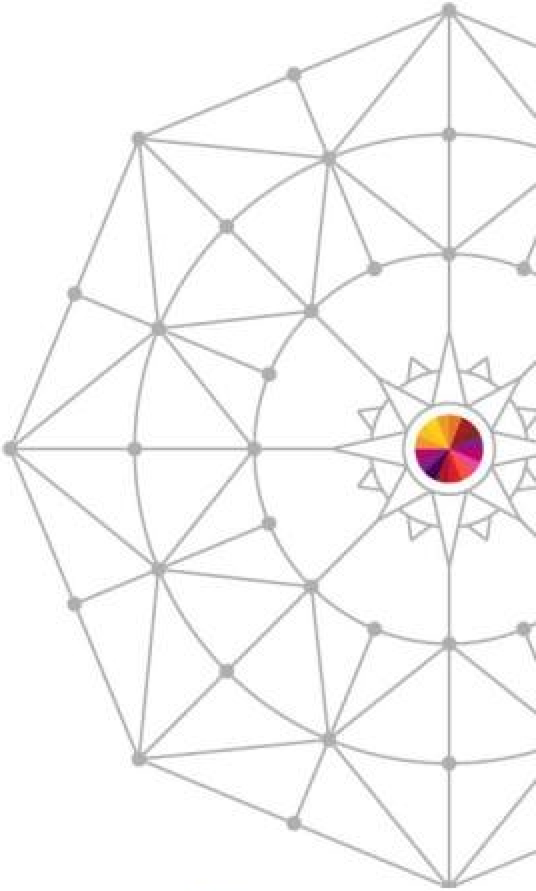
?

Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “ Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Notes

- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
- This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.



TPFUG Washington, D.C.