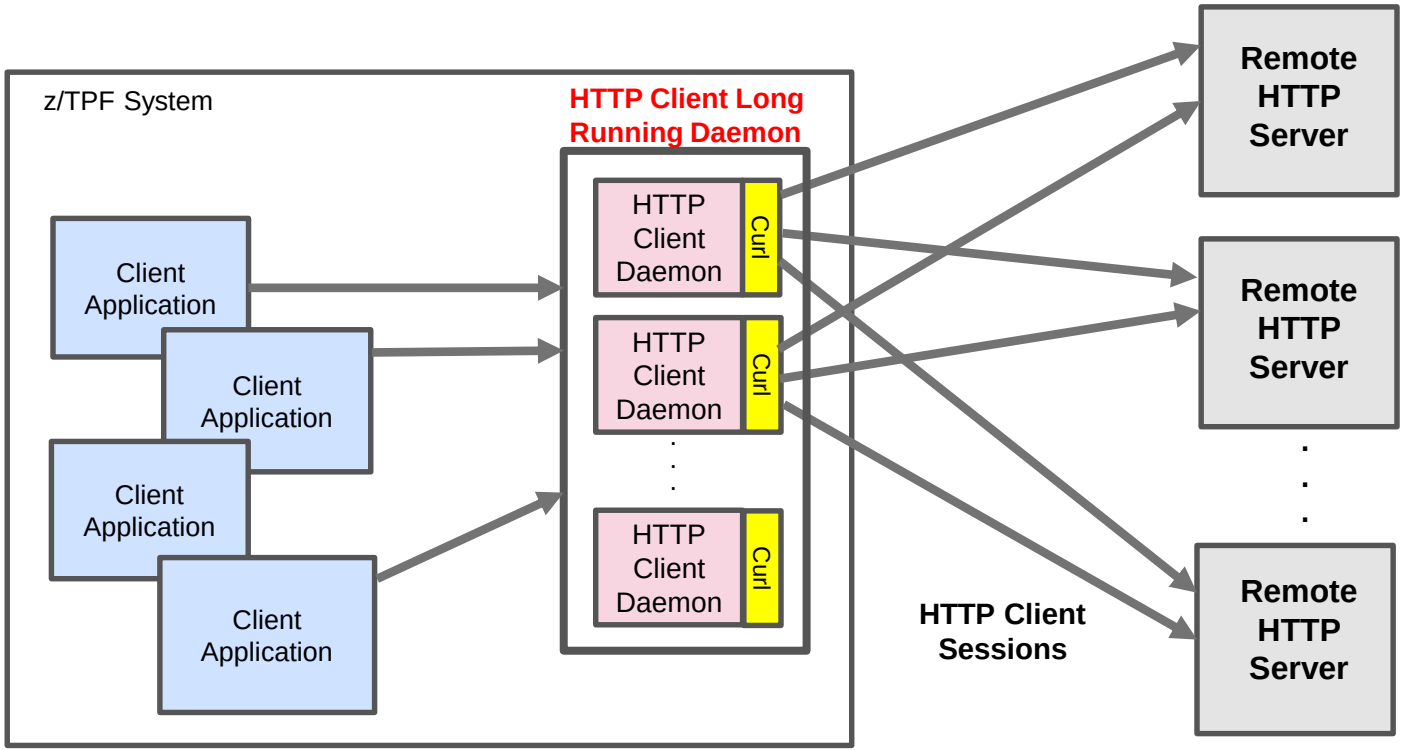
z/TPF Enhanced HTTP Client and High Speed Connector Enhancements

Jamie Farmer
Communications Subcommittee

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.

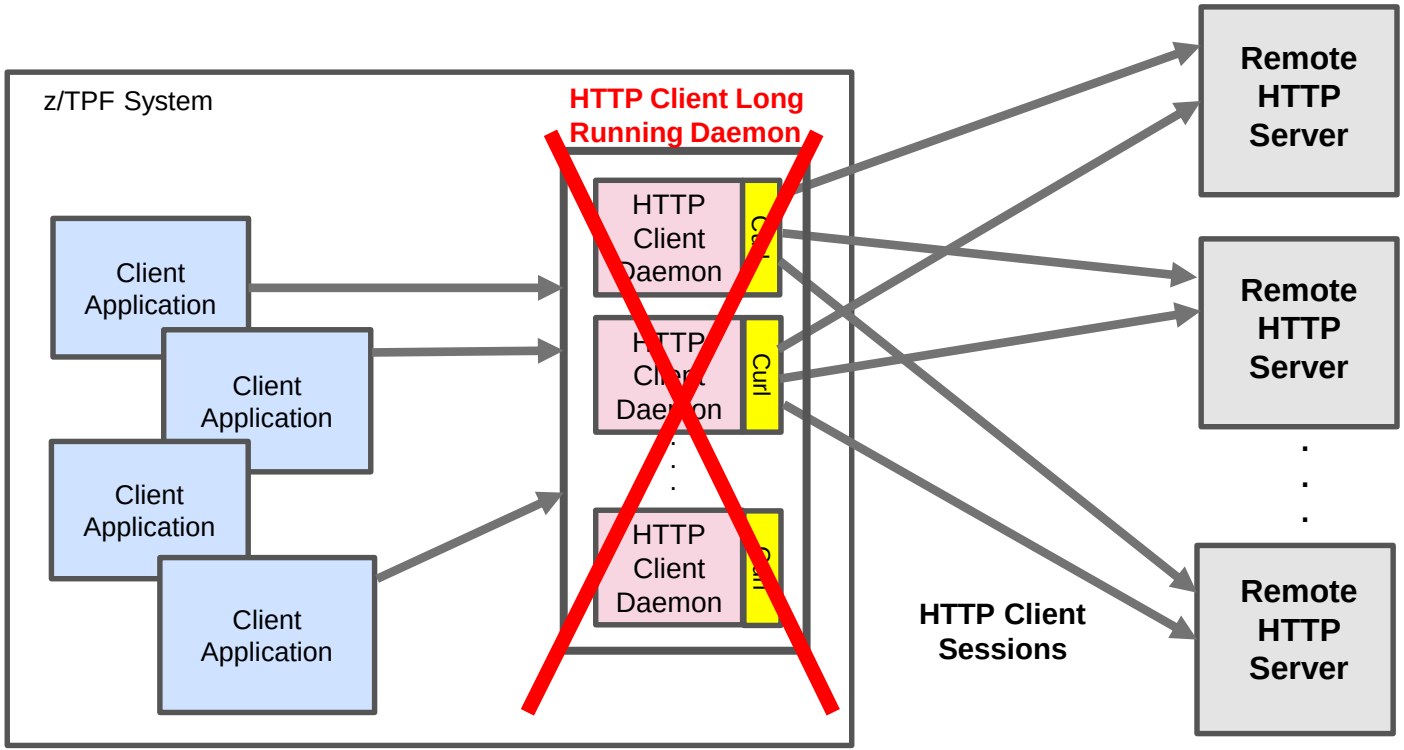
Original HTTP Client Support



Problems with Original HTTP Client

- Long Running Daemons are not automatically recycled.
- Inefficient model to pass data across processes.
- Current libCurl package does not support the latest TLS standards.
- REST consumer would require significant changes to original HTTP client support.

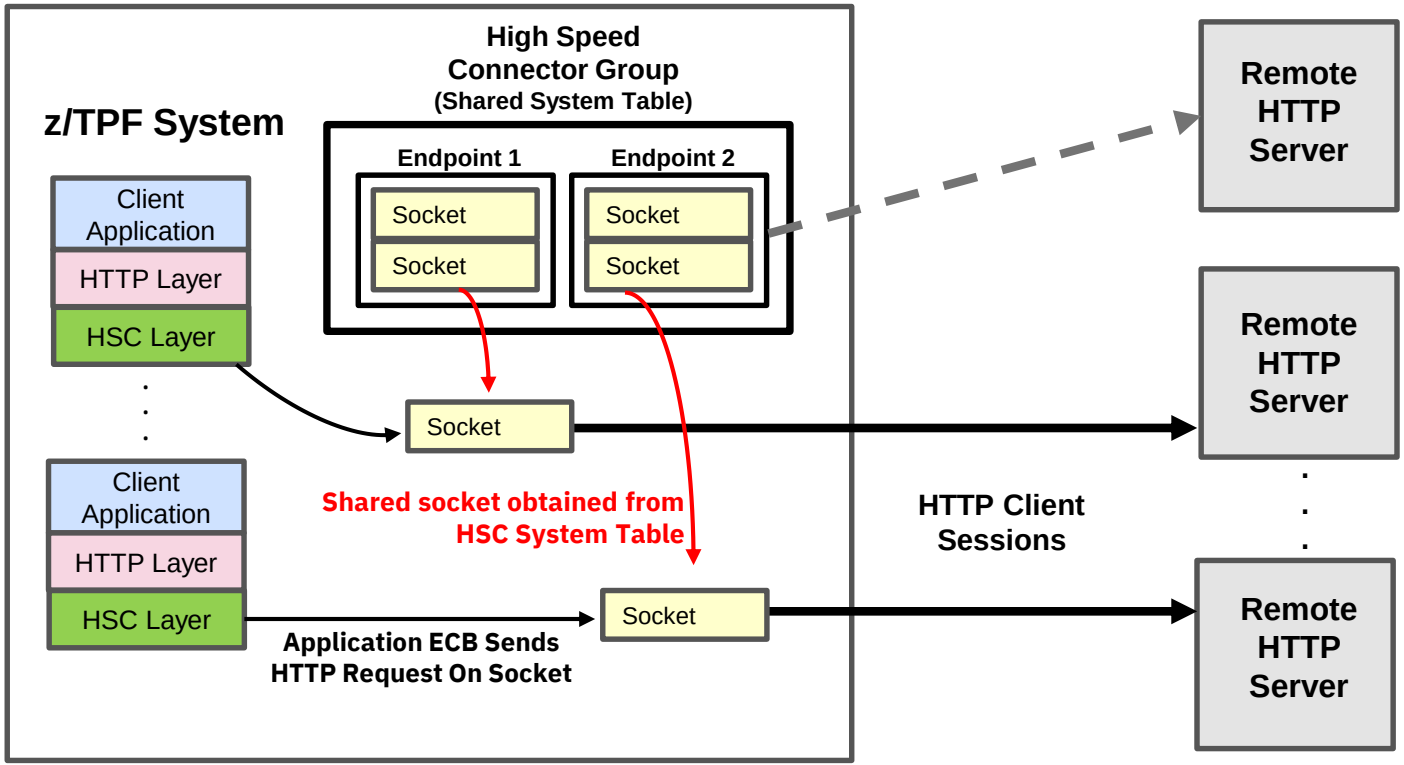
Original HTTP Client Support



Problems with Original HTTP Client

- Long Running Daemons are not automatically recycled.
- Inefficient model to pass data across processes.
- Current libCurl package does not support the latest TLS standards.
- REST consumer would require significant changes to original HTTP client support.

Enhanced HTTP Client



Enhanced HTTP Client

- No long running daemons
- No longer passing data across processes
- Leveraging high speed connector

HTTP Client Terminology

HTTP Client Session Types

HTTP Client Persistent Sessions

Long running HTTP client sessions that can be shared by any z/TPF application ECBs and managed by the z/TPF high speed connector.

HTTP Client Non-Persistent Sessions

HTTP client session that is established and subsequently torn down after processing a single request.

HTTP Client API Requests

Synchronous HTTP Client Requests

The application ECB issuing the HTTP client request does not receive control back until the response is received (or a timeout occurs)

Asynchronous HTTP Client Requests

The application ECB issuing the HTTP client request can exit and a new application ECB is created when the response is received.

tpf_httpSendRequest Format

```
LIBS := CHTE
#include <tpf/c_https.h>

int tpf_httpSendRequest(char *host, t_httpClientRequest *requestParms,
                        tpf_httpsvr_resp **response, t_httpClientConnect *connectParms,
                        int options);
```

host: Host or IP address to establish session to
requestParms: A t_httpClientRequest structure containing the HTTP client request information
- **HTTP Version:** only supports HTTP 1.1
- **Request Type:** GET, HEAD, PUT, POST, DELETE
- **uri:** The service on a host to be accessed
- **timeout:** How long to wait to send request and receive response (in milliseconds)
- **headers:** User defined headers to include in request
- **body:** User defined body to send to remote (Ignored if not PUT / POST)

response: Pointer to an tpf_httpsvr_resp structure containing the response
connectParms: Non-persistent connect options (ignored for persistent sessions)
options: Not used – will be used for future extensions

tpf_httpSendAsyncRequest Format

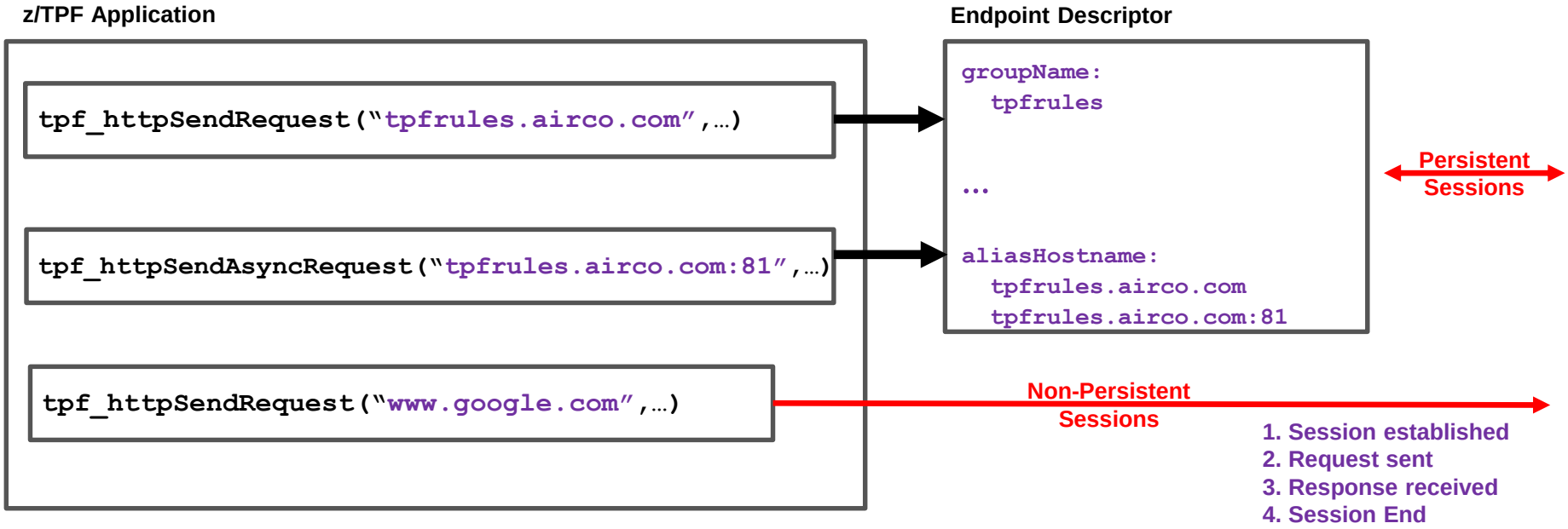
```
LIBS := CHTE
#include <tpf/c_https.h>

int tpf_httpSendAsyncRequest(char *host, t_httpClientRequest *requestParms,
                        tpf_httpsvr_resp **response, t_asyncRequestParms *asyncParms,
                        t_httpClientConnect *connectParms,  int options);
```

asyncParms: The asynchronous parameters for the request containing the program to invoke when the response is received as well as user data to pass.

options: TPF_HTTP_KEEP_REQUEST option to save original HTTP request across ECBs during the asynchronous application call

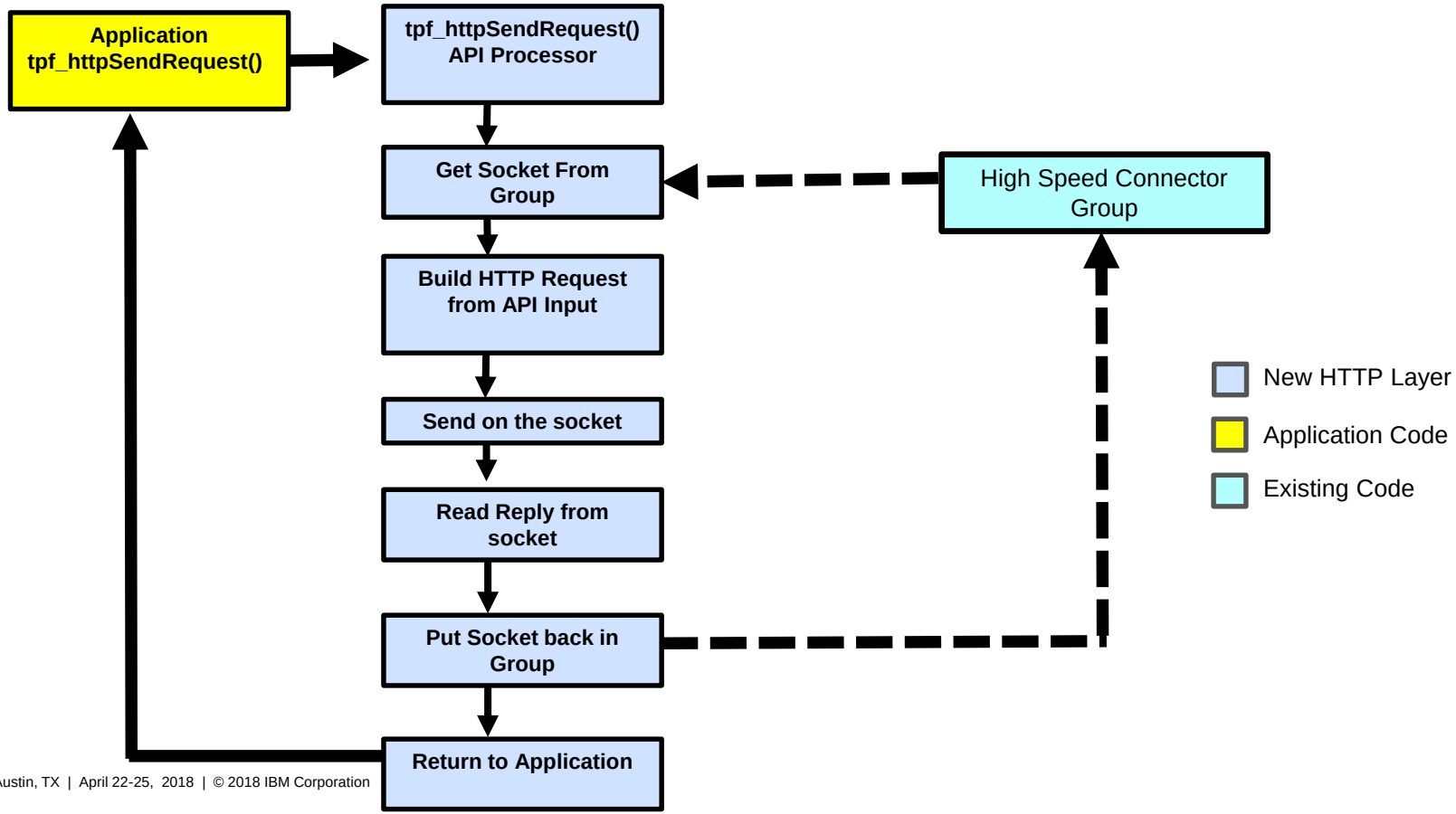
Persistent vs Non-Persistent Sessions



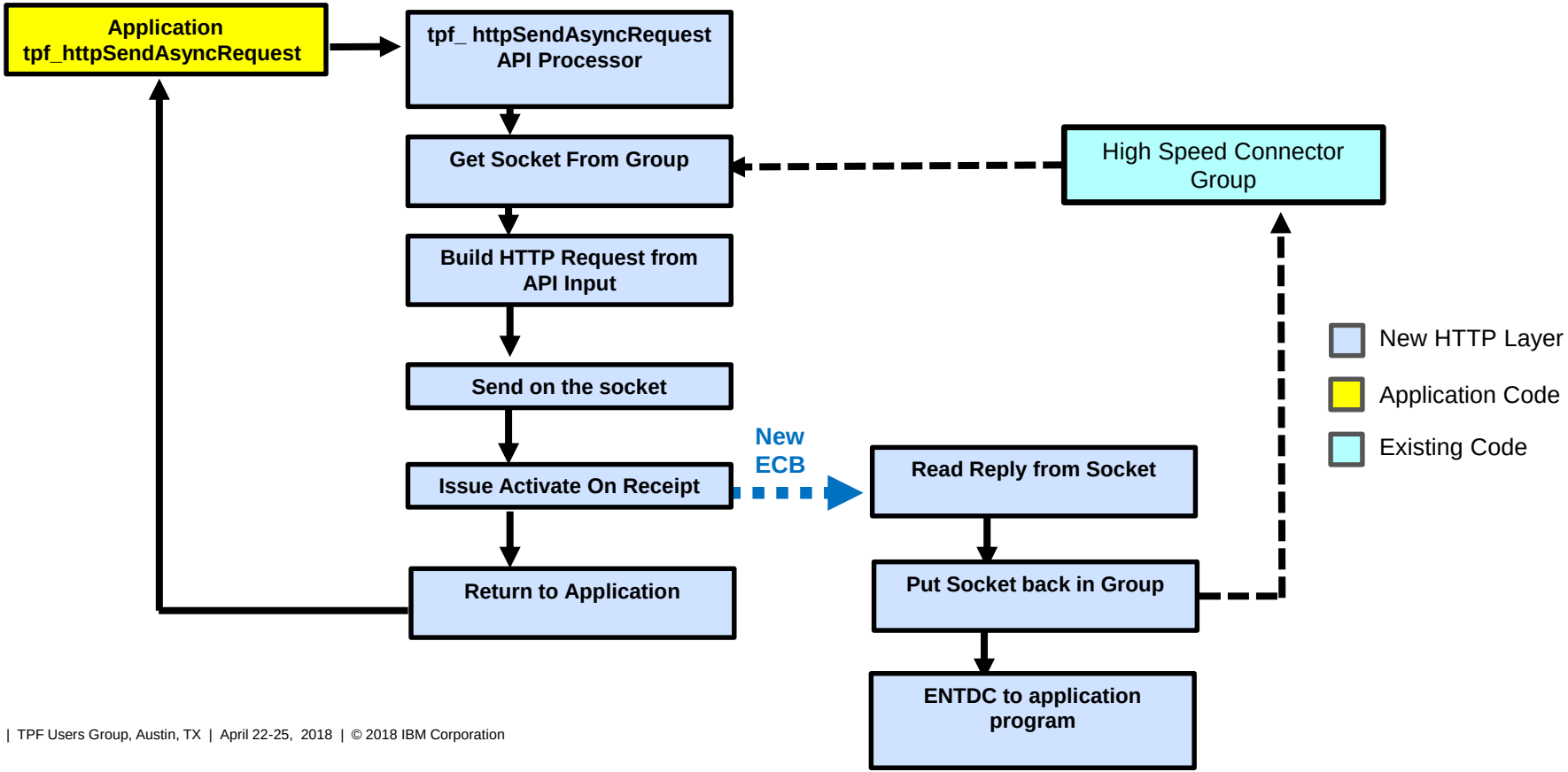
Persistent vs Non-Persistent sessions are transparent to the application

- Allows an administrator switch from non-persistent to persistent sessions with no application changes!

Synchronous HTTP Client Request For Persistent Sessions



Asynchronous HTTP Client Request For Persistent Sessions



Defining Persistent Sessions

```
<tns:groupName>tpfrules</tns:groupName>
 <tns:groupType>HTTP</tns:groupType>
 <tns:qMaxDepth>100</tns:qMaxDepth>
 <tns:qThreshold>80</tns:qThreshold>
 <tns:syncTimeout>200</tns:syncTimeout>
 <tns:maxAsyncData>10000</tns:maxAsyncData>
 <tns:maxAsyncMessage>10000</tns:maxAsyncMessage>
 <tns:aliasHostname>tpfrules.airco.com</tns:aliasHostname>
 <tns:aliasHostname>tpfrules.airco.com:81</tns:aliasHostname>
          .
          .
          .
 <tns:Endpoint>
   <tns:endpointName>httprulp</tns:endpointName>
   <tns:role>PRIMARY</tns:role>
   <tns:destination>httprulp.airco.com</tns:destination>
   <tns:startSocket>10</tns:startSocket>
   <tns:maxSocket>20</tns:maxSocket>
   <tns:bufferSendSize>262144</tns:bufferSendSize>
   <tns:bufferReceiveSize>262144</tns:bufferReceiveSize>
 </tns:Endpoint>
```

Endpoint Descriptor File

Load – Version Control File System

Changing the Persistent Session Configuration

- Transparent to the z/TPF applications
  - Increase capacity (adding more endpoints/socket)
  - Change endpoint definitions (ie. socket buffer sizes)
  - Change group definitions (queue sizes, timeouts, thresholds)

- Simply load an updated version of the file through the z/TPF loader package to apply updates
  - **Changes take effect immediately for existing endpoint groups!**

What If I’m Using the Original HTTP Client Today?

- This is not a rip and replace!
- New HTTP client APIs have been created for use with the enhanced HTTP client
- The two supports can be run concurrently
  - Allowing for a controlled migration of your applications to the new support

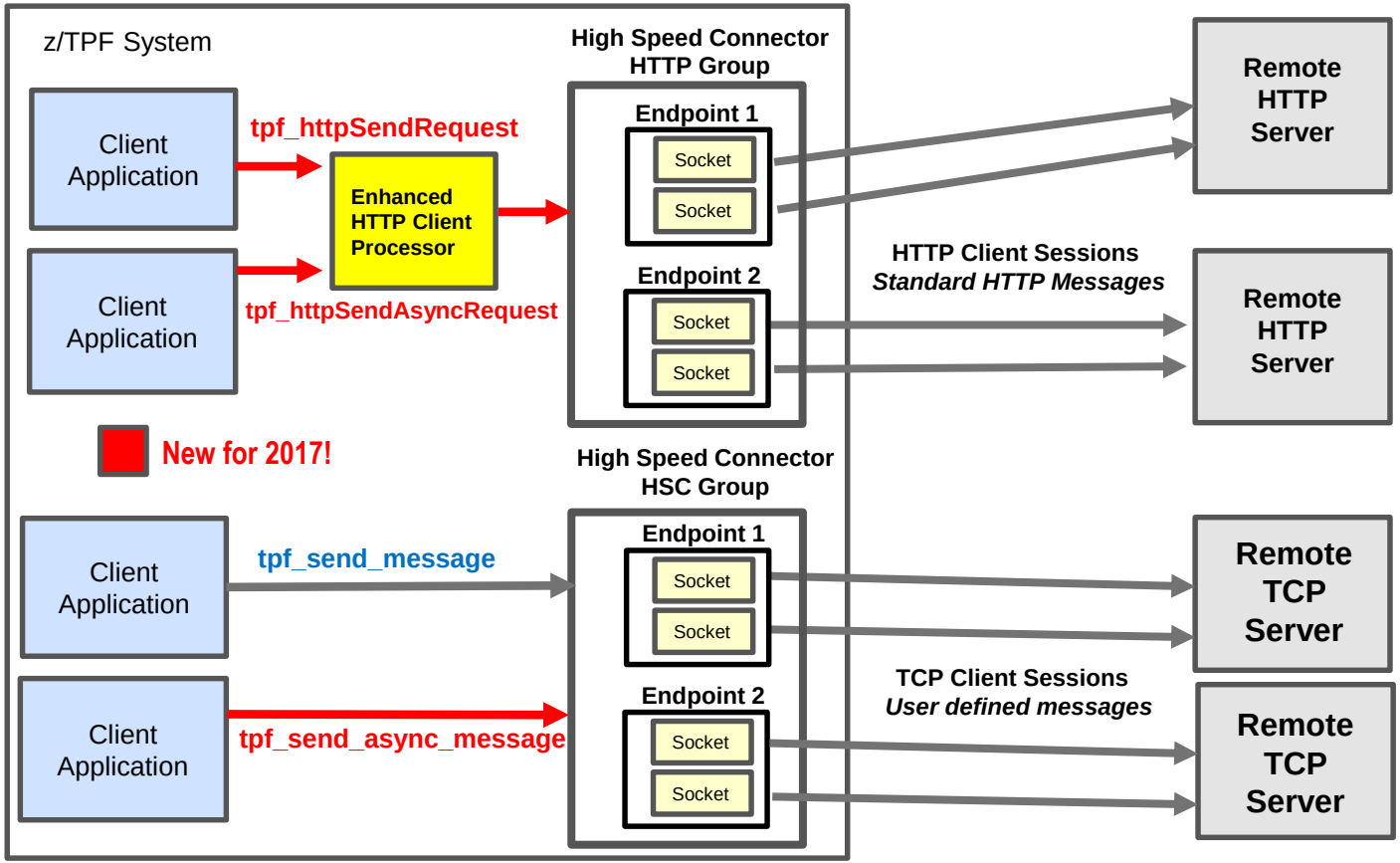
Enhanced HTTP Client Performance

- Up to **80% improvement** in performance using the Enhanced HTTP Client vs the existing HTTP Client
  - Optimized socket processing
  - Cross process request/reply copies not required

Enhanced HTTP Client Deliverable

- APAR PJ44733 was delivered
  - December 2017 - PUT 14
- High Speed Connector and Enhanced HTTP is 100% TE-Eligible

High Speed Connector



High Speed Connector

- z/TPF managed persistent client sessions
- Ability to send standard HTTP messages
- Ability to send user defined messages asynchronously

Asynchronous High Speed Connector API For User Defined Messages

- New tpf_send_async_message function.
- Same parameters as existing tpf_send_message API, except for the asynchronous information
  - Ability to specify the program to enter in new ECB when response is received
  - Ability to send user data to the new application ECB
- Included in APAR PJ44733 delivered in December 2017 – PUT 14

What's Coming?

- Actively working on enabling high speed connector and Enhanced HTTP Client with Transport Layer Security (TLS)
  - Latest TLS standards will be supported

- Actively looking at enabling Apache-2 with the latest TLS standards.
  - z/TPF HTTP Server already supports latest TLS standards

Summary – PJ44733

- Provides a better performing HTTP Client
  - Persistent and non-persistent sessions
  - Synchronous / Asynchronous application HTTP client APIs
- Provides infrastructure for z/TPF REST consumer support
- Provides new asynchronous application API for high speed connector
- Improved dynamic management of high speed connector groups

Thank You!

Questions?

Trademarks

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.