Spectrum Scale User Days
19.3 – 21-3. 2019

Session: Architekturen und Use Cases (nicht nur für Z)

—

Gerd Grundke
Manager Spectrum Scale on Z & Core

IBM

Inhalt

Die Sprecher

Gerd Grundke

IBM Research and Development

Head of Software Development for Spectrum Scale on Z

Mail: gerd.grundke@de.ibm.com

Stefan Amann

IBM Research and Development

STSM, zSystems storage architect

Mail: stefan.amann@de.ibm.com

Christian May

IBM Research and Development

Spectrum Scale for Linux on Z, Client Enablement

Mail: christian_may@de.ibm.com

Susanne Wintenberger

IBM Research and Development, Client Center EMEA

Z ATS (Advanced Technical Skills)

Mail: swinten@de.ibm.com

Andreas Luengen

IBM Research and Development,

IT Specialist, Proactive Service Developer

Mail: luengen@de.ibm.com

Praxisbeispiel:
# Design Konzept z14 und SSC mit Spectrum Scale

—

Stefan Amann
STSM, zSystems storage architect

Driving the future of IBM Z

1. Delivering differentiated client value via continuous innovation in hardware and software technology

2. Growing IBM Z in a world where the dominant IT consumption model is cloud services

3. Expanding IBM Z capabilities to support modern digital workloads
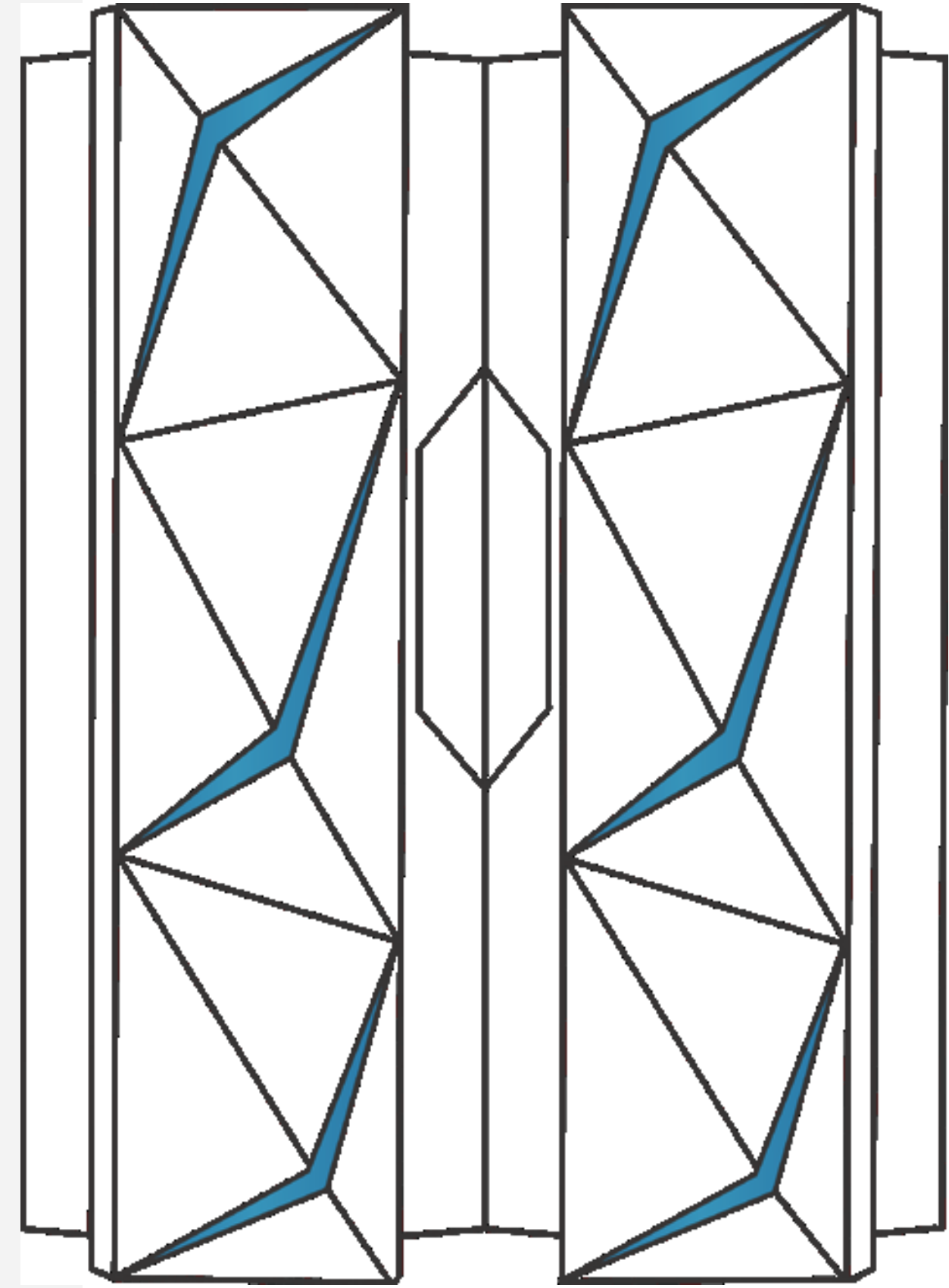
IBM z14
Machine Type: 3906
Models:
M01, M02, M03, M04, M05

Model	Customer PUs	Max Memory
M05	170	32 TB
M04	141	32 TB
M03	105	24 TB
M02	69	16 TB
M01	33	8 TB
ZR1	4, 12, 24, 30	8 TB

IBM z14
Machine Type: 3907
Model ZR1

- Up to 170 Customer PUs
  - Equivalent to ~1300x86 cores
- Up to 32 TB memory
- Up to 85 LPARs
  - up to 16 TB per LPAR
- I/O requirements up to 9 million IOPS, raw I/O bandwidth of 832 GB/S

IBM LinuxONE Emperor™ II
Machine Type: 3906
Models:
LM1, LM2, LM3, LM4, LM5

Model	Customer PUs	Max Memory
LM5	170	32 TB
LM4	141	32 TB
LM3	105	24 TB
LM2	69	16 TB
LM1	33	8 TB
LR1	4, 12, 24, 30	8 TB

IBM LinuxONE Rockhopper II
Machine Type: 3907
Model LR1

Z14 – Designed for trusted digital experiences
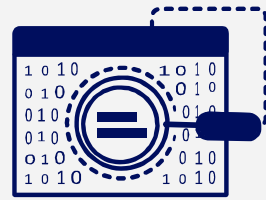
Secure 100% of data
through
pervasive encryption

Gain new insights
with
IBM Machine Learning

Speed digital innovation
through open and
connected cloud

IBM Secure Service Container Core Values and Offerings
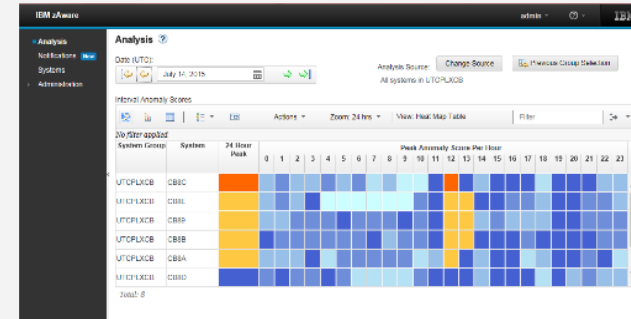
On-Premise Offerings

IBM DB2 Analytics Accelerator for z/OS

Accelerated DB2 Queries

IBM z/VSE Network Appliance

Provides TCP/IP network access without requiring a TCP/IP stack in z/VSE

IBM Operations Analytics for z Systems (zAware)

IT Analytics Anomaly Detection

Secure Service Container for IBM Cloud Private

Secure deployment of Docker / Kubernetes based Solutions to on-premise IBM Z or IBM LinuxONE Hybrid / Private Clouds

IBM Hyper Protect Services

- *IBM Cloud Hyper Protect Crypto Services*
- *IBM Cloud Hyper Protect Containers*
- *IBM Cloud Hyper Protect DBaaS*
- *IBM Cloud Hyper Protect Virtual Servers*

Security
- Protection from misuse of privileged user credentials
- Solution can leverage appliance security features without code changes
- Only boots untampered appliances
- Data and code is encrypted in flight and at rest
- No access to memory or processor state
- EAL5+ LPAR isolation of each solution environment
- No direct host or OS level interaction (REST APIs)

Deployment
- Deploy solution instead of different components
- Deploy appliance without needing Operating System skills

**Management**
- Manage appliance without needing Operating System skills
- Limited variance of settings
- Automation via RESTful APIs
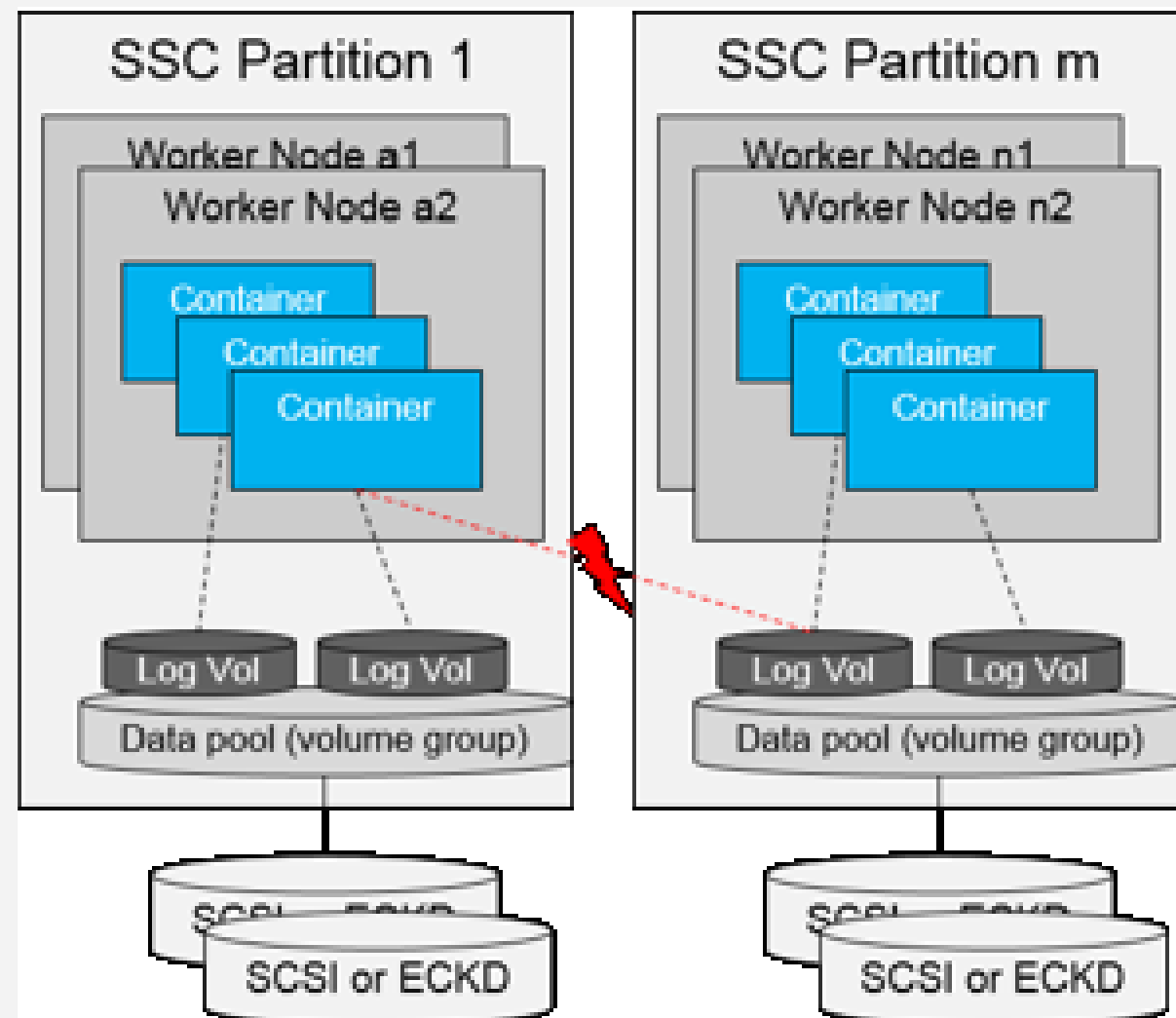- UI for better user experience

IBM Secure Service Container Appliance Concept

IBM Secure Service Container Appliance

Management UI / REST API

Solution / Application

Application Interfaces

Management Backend

Base Operating System
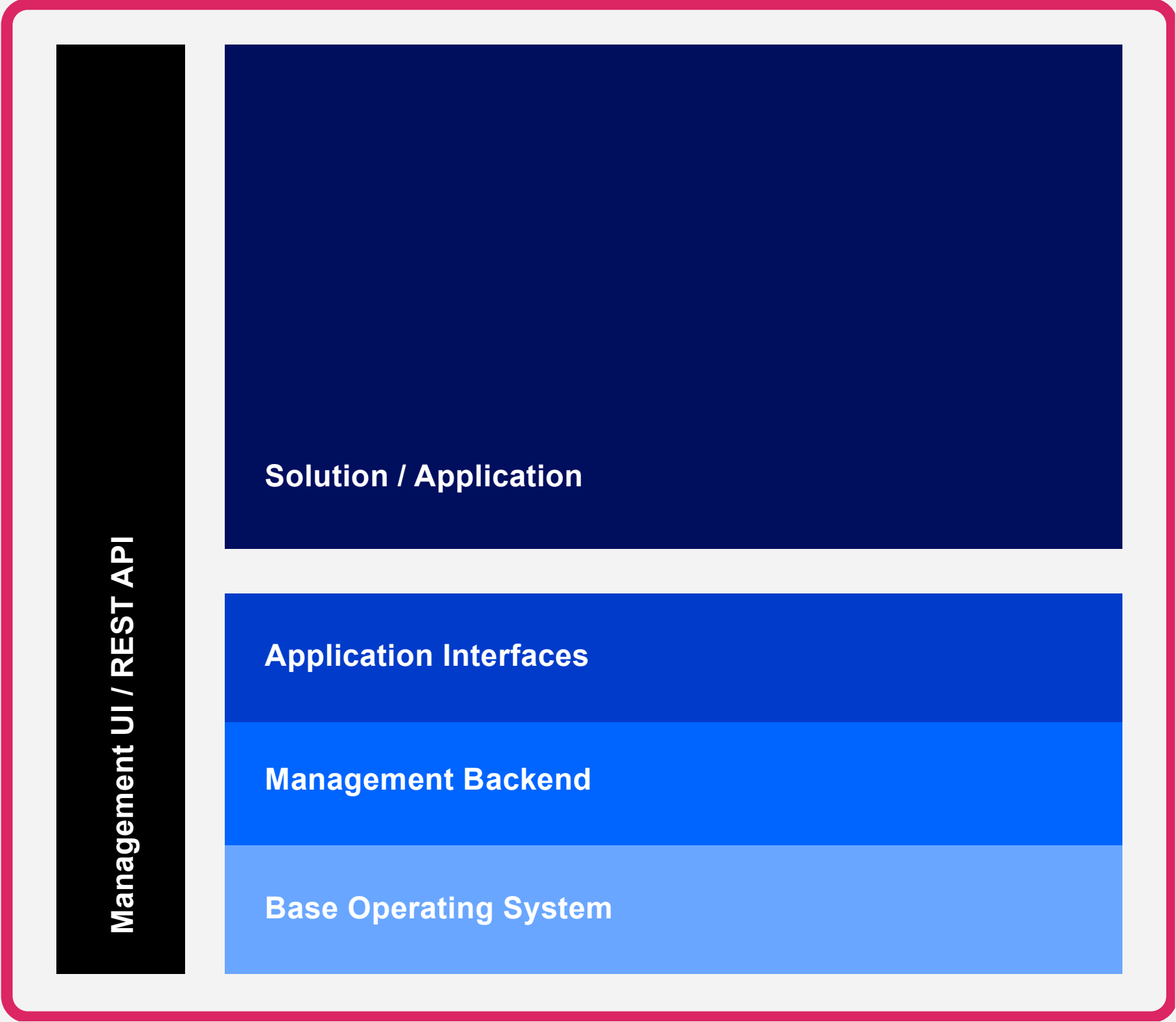
# IBM Secure Service Container and Storage

Dedicated Storage

Drawback of running with dedicated storage

- IBM Secure Service Container based Applications run in LPARs

  - Similar to virtual machines

- Storage volume are dedicated to single LPARs

- Concept does not allow workload to move to other LPARs

  - Maintenance, or LPAR failures

IBM Secure Service Container and Shared Storage

Shared Storage
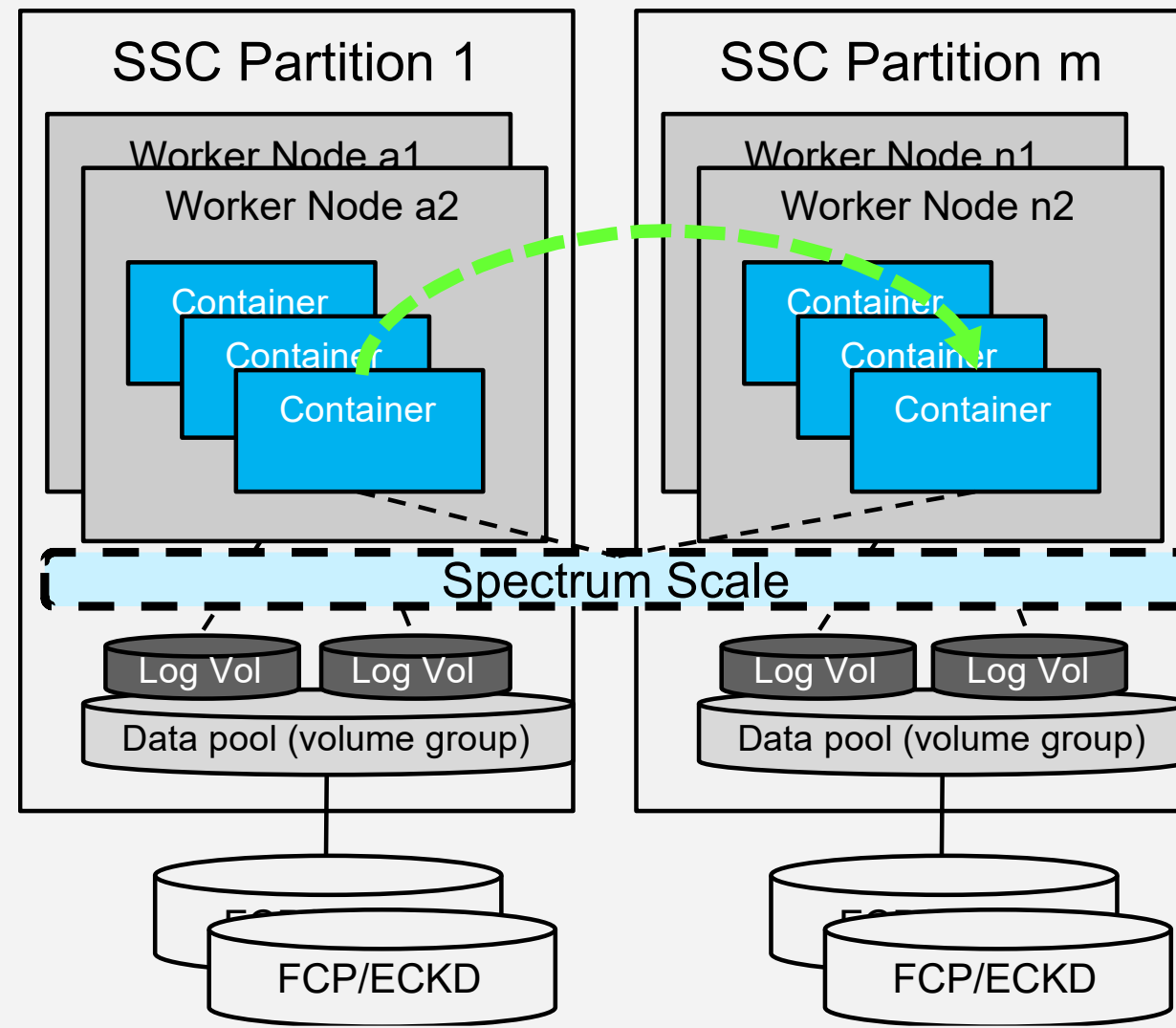
Spectrum Scale provides cluster file system across worker nodes

- Provides persistent data volumes
- May span multiple partitions / systems
- Allows workloads to be started on different worker nodes
  - For example, when a system / partition requires service
- Replicates data

Goals

- seemless integration of Ubiquity/Spectrum Scale
- automated provisioning of a cluster

New Technology Introduction:
IBM Systems Early Programs

—

Christian May
Spectrum Scale for Linux on Z,
Client Enablement

IBM Systems Early Programs

What is an Early Client Program?

❖ An Early Client Program is a partnership between IBM and Clients with advantages for all.
- IBM provides early hardware (HW), software (SW) or both to Clients for validation in their environments.
- Clients install, use, validate and then provide feedback on the early product.

❖ Nomination surveys and selection criteria are used

❖ Clients accept program contract

❖ IBM provides early code, documentation, support, NTI (New Technology Introduction) rep, access to subject matter experts (SME), validation guidance, education, forum, and problem assistance*

❖ Program requirements can vary but include function validation, production experience, and named references*

❖ NTI reps collect regular status, run regular calls with clients, provide assistance as needed*

*Note: Not all products offer all program types and some products offer additional programs.

IBM Systems Early Programs

New Technology Introduction Program Examples:

- ❖ IBM Spectrum Storage Solutions (Software Defined Storage)

- ❖ IBM Elastic Storage Server (ESS)

- ❖ IBM System Storage DS8000 (High-End Storage)

- ❖ IBM Flash Systems A9000 (Mid-Range Storage)

- ❖ IBM Cognitive Systems (POWER Systems and Software)

- ❖ IBM Z (IBM z14), IBM LinuxONE

- ❖ Operating Systems (IBM z/OS)

IBM Spectrum Scale

IBM Systems Early Programs

Early Program Types:

	Beta Programm	Early Support Programm (ESP)	Early Acceptance Programm (EAP)
Product Level	Early development and test phase (early build)	Final development phase (later build)	General Availability (GA)
Focus	Early evaluation of functionality and usability	Evaluation of complete GA offering	Early adoption of new product with smooth transition into production
Timing	Early test phase	Later test phase	GA
Support	IBM Development	IBM Development and IBM Support	IBM Support
Exit	Under development and test	GA level service provided	GA+ level service as appropriate

IBM Systems Early Programs

Advantages for Clients:

❖ Early Access to SW, HW, or both

❖ Early Information Access and Education

❖ Access to Development

❖ Early Program Representative

❖ Critical ISV (Independent Software Vendors) Assistance

❖ Problem Escalation

❖ Opportunity to Influence Product Design

❖ Strengthen Partnership with IBM

❖ Competitive Edge

IBM Systems Early Programs

Advantages for IBM:

- ❖ Obtain Client Feedback and future requirements

- ❖ Validate GA Readiness in Client environments

- ❖ Improve Product Quality at GA

- ❖ Obtain Named References

- ❖ Improve Client Relationship (Partnering/Loyalty)

- ❖ Increase Client Satisfaction

- ❖ Pipeline of Potential Clients

- ❖ Validation of many aspects of the product

IBM Systems Early Programs

Program Requirements*

❖ Validate top solutions, features or functionality

❖ Run in test, production or production-like environment

❖ Provide references

❖ Commit resources to meet program objectives

❖ Provide regular status and feedback

❖ Participate in regular calls

❖ Roll out regular service updates

*Note:Program requirements vary by program and product, but might include the following items:

IBM Systems Early Programs

More Information and Program Contacts

❖ IBM Early Programs: https://ibm.biz/NTIPrograms
❖ IBM Systems: www.ibm.com/it-infrastructure/servers
❖ IBM Storage: https://www.ibm.com/it-infrastructure/storage
❖ IBM Spectrum Storage: https://www.ibm.com/it-infrastructure/storage/spectrum

❖ Contact an NTI Program Manager for Storage or referral to any program type:
  ❖ Ben Rawlins (bkrawlin@us.ibm.com)
  ❖ Jodi Everdon (jeverdon@us.ibm.com)
  ❖ Linda Muller (lmuller@us.ibm.com)
  ❖ Mary Anne Filosa (mfilosa@us.ibm.com)
  ❖ Marie Romero (mlestrad@us.ibm.com)

❖ NTI Program Interest Form: www-355.ibm.com/technologyconnect/cna/epInterestForm.xhtml

# Praxisbeispiel:
Cross Platform Cluster

—

Susanne Wintenberger
Z ATS (Advanced Technical Skills)

IBM

Who they are / What they do

- International banking and financial services company
  - A large amount of accounting is done in z/OS
  - Nightly Java batch processes create extract files, that are then sent to the Distributed DataStage Spectrum Scale clusters for further processing.
  - In addition other files may be created on DataStage that needs to be transferred back to IBM Z.
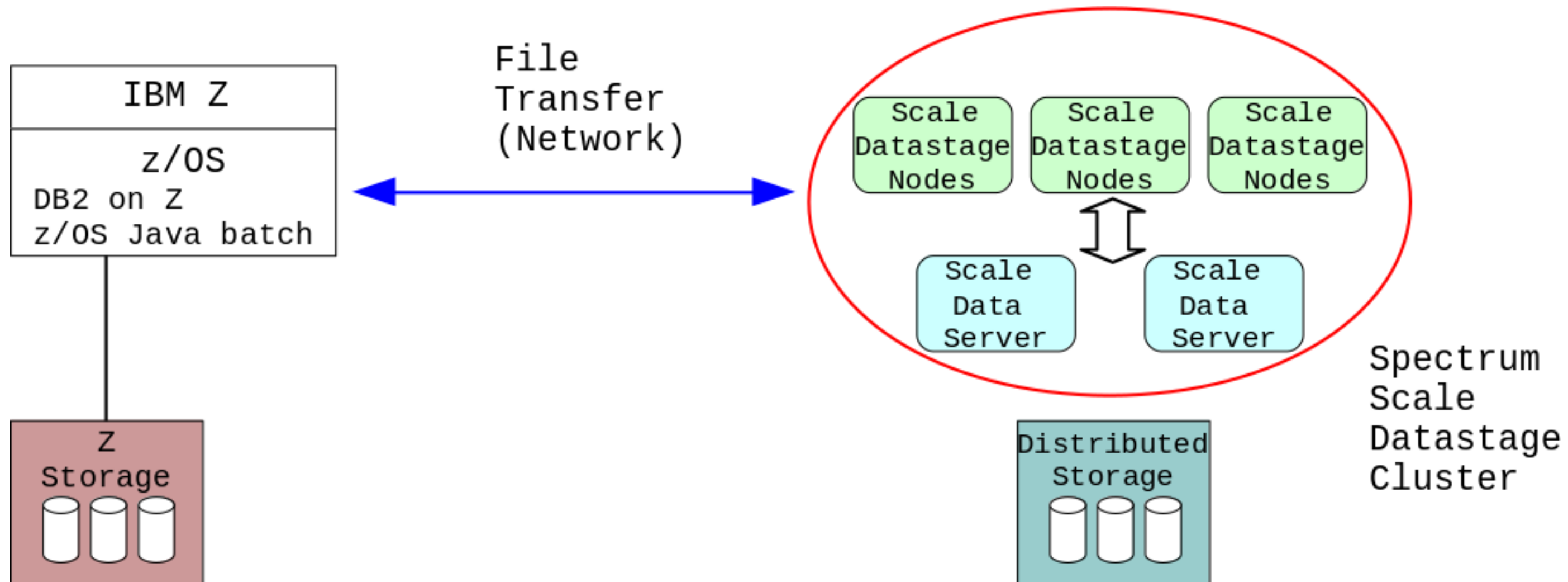
Who they are / What they do

- Interest/Requirement:
  - Leverage IBM Spectrum Scale to integrate data across heterogenous hardware solutions
  - Integrate in a flexible and resilient way
  - Deliver **near real time access to data for Analytics**

- Goal:
  - Look for a software defined storage solution to provide high performance global namespace

- Started PoC in summer 2018
  - Evaluate various scenarios to integrate data across different operating systems and/or different platforms

Who they are / What they do
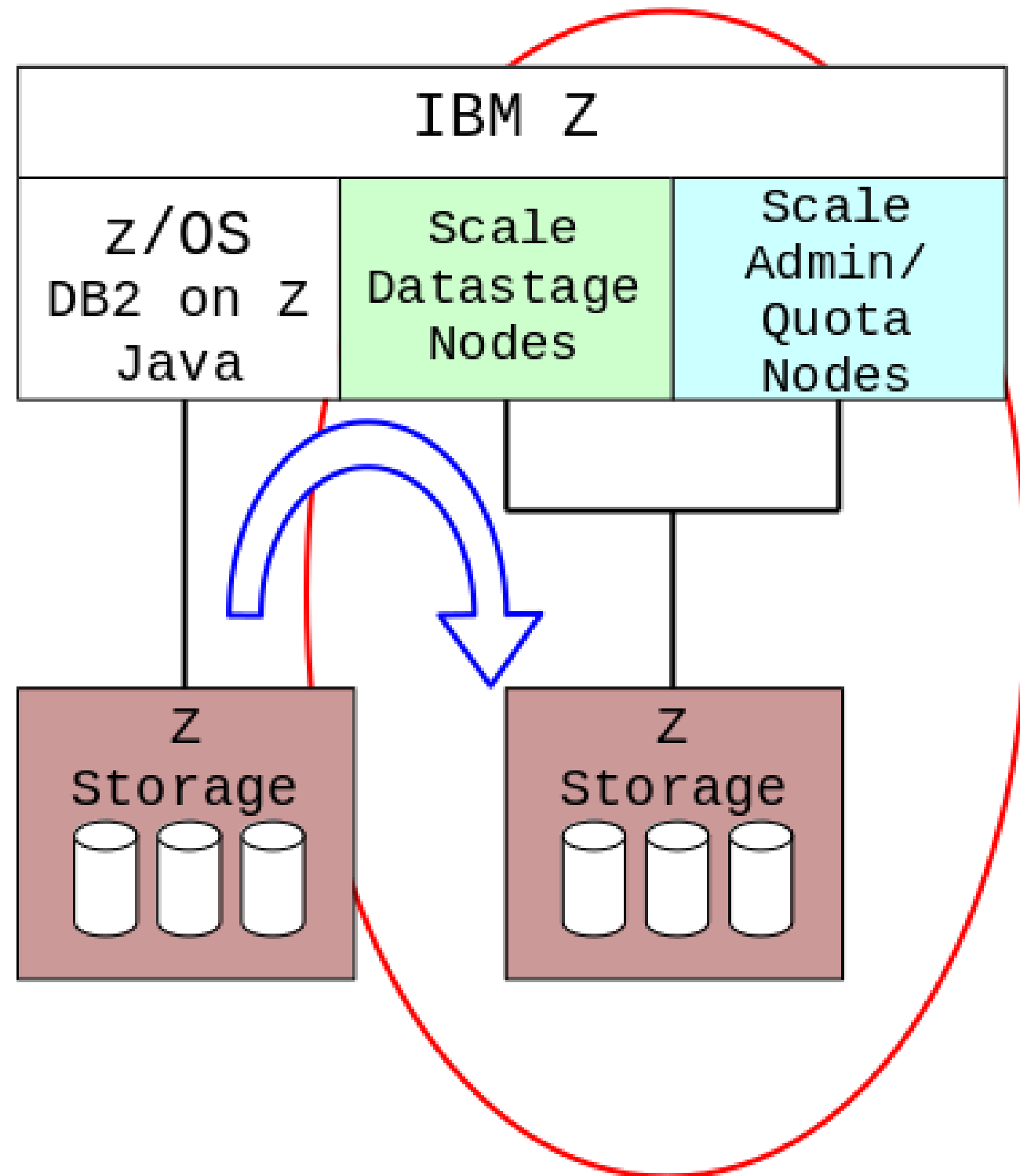
- Scenario 1a

  - Run Spectrum Scale Datastage cluster on IBM Z

  - Leverage z/OS and Linux proximity on the same IBM Z (between z/OS and Datastage/Spectrum Scale)

  - Evaluate better data currency due to shorter data lag with file transfer over OSA or Hipersockets on the same CEC

Who they are / What they do
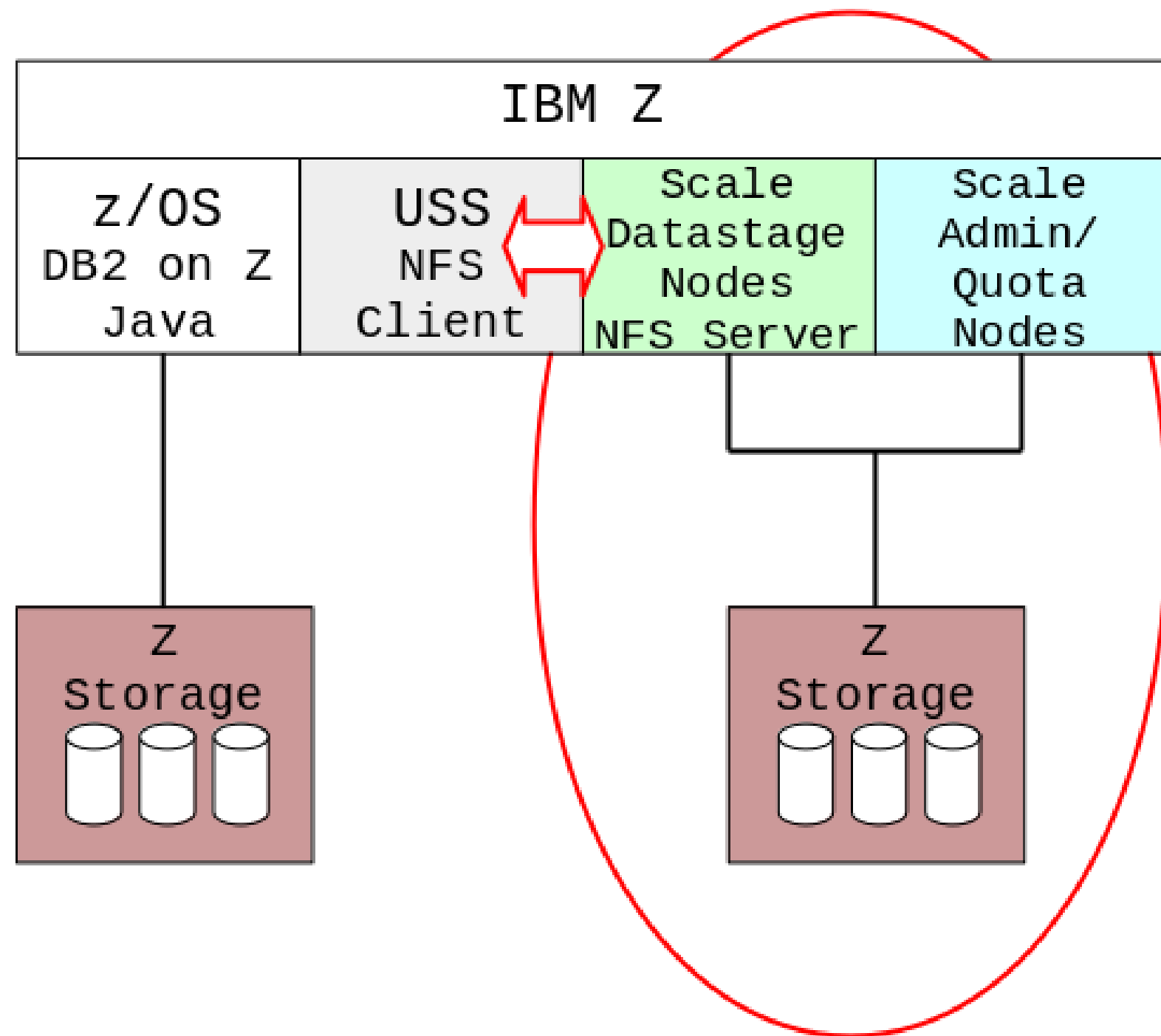
- Scenario 1b

  - Run Spectrum Scale Datastage cluster on IBM Z

  - NFS client-server connection over TCP/IP using Hipersockets or OSA-Express physical connection

  - Mount Linux on Z Spectrum Scale directory on z/OS Unix System Services file system

 - "Looks like" local z/OS USS file system directory to z/OS Java Batch

Who they are / What they do

- Scenario 2b
  - Use Spectrum Scale remote mount feature between Linux on Z cluster and existing Spectrum Scale Datastage cluster
  - Evaluate the Spectrum Scale's "cross-mount" capability and thus the huge parallel performance to and from Z mainframe to distributed
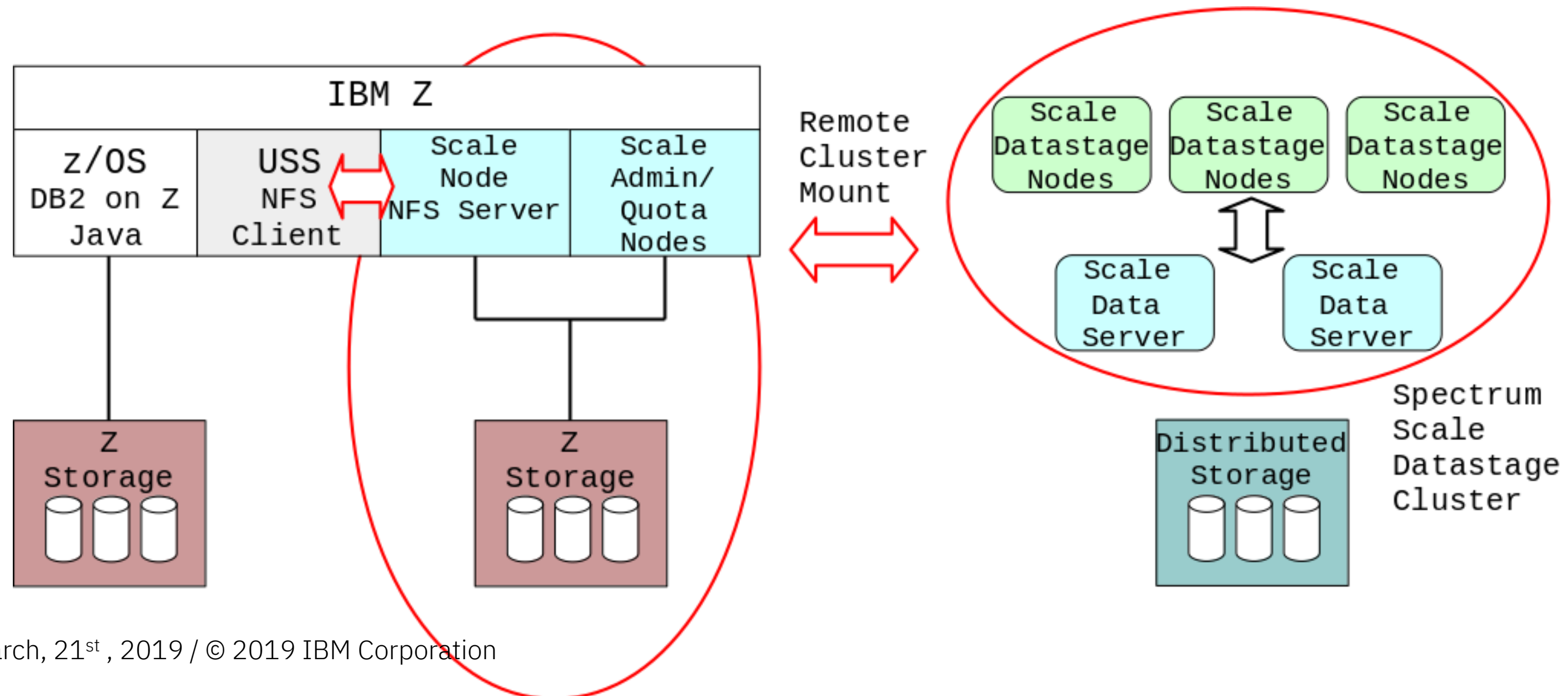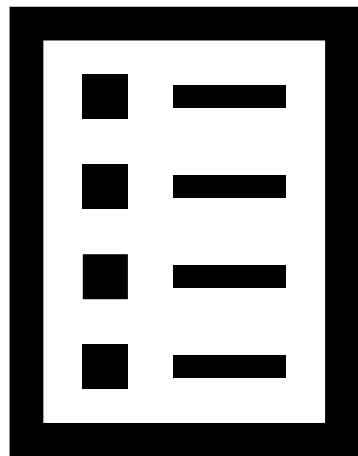
Praxisbeispiel:
Call Home

—

Andreas Luengen
IT Specialist, Developer
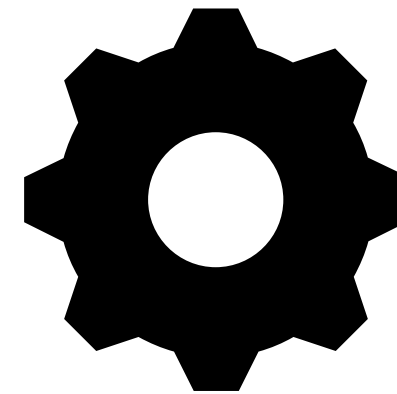Spectrum Scale – Proactive Service
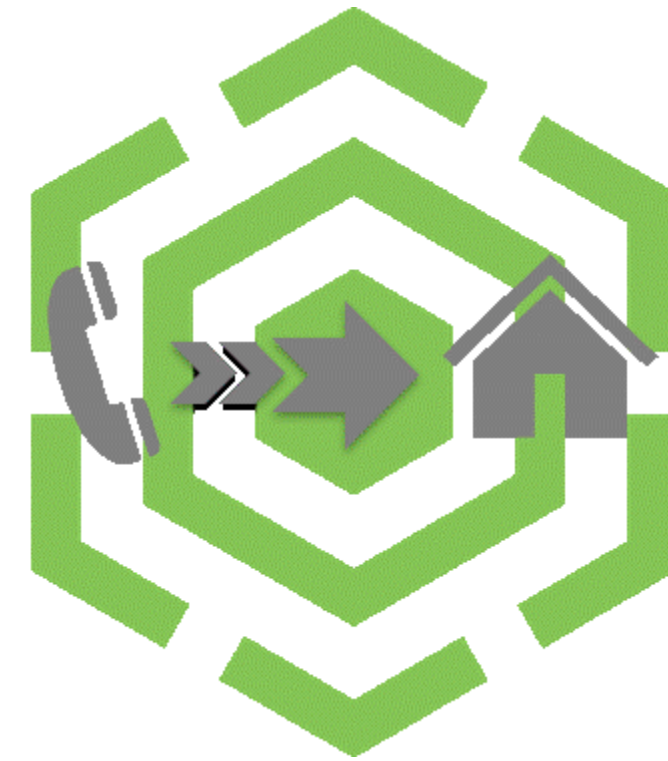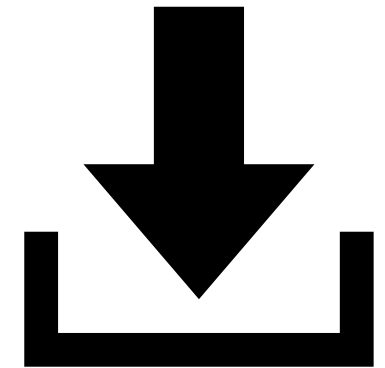
Call Home

Overview

Why Use

Config

Collect

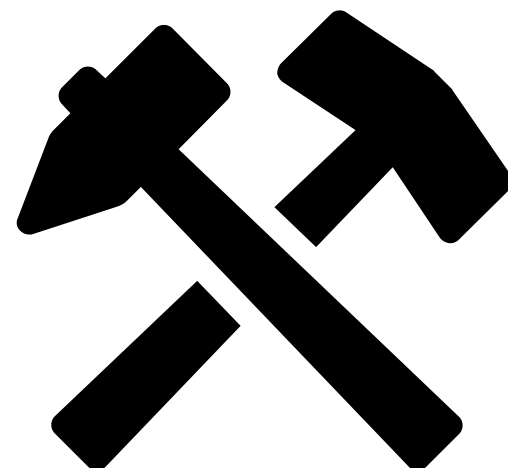**What's new**
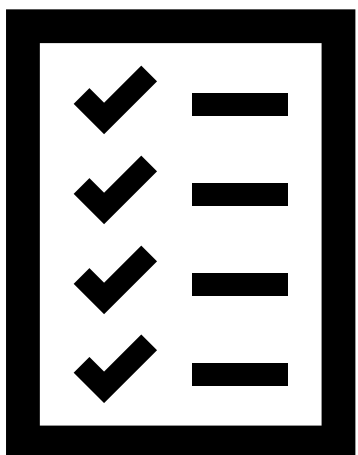
Usage
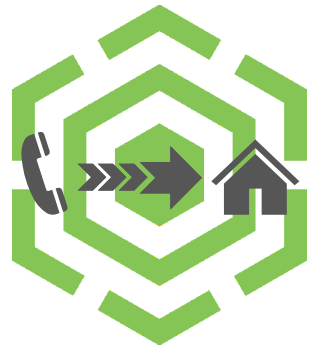
Cheat
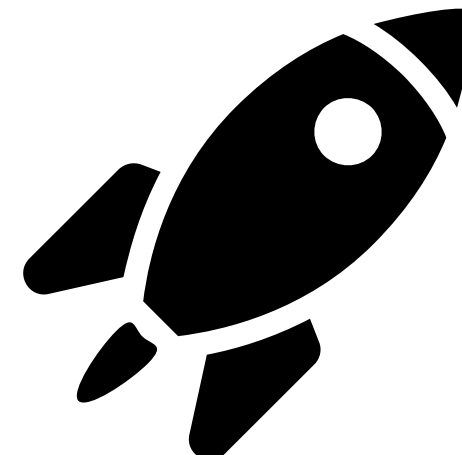
Done
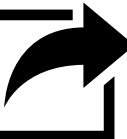
Software Call Home Architecture (today)

What is it used for?
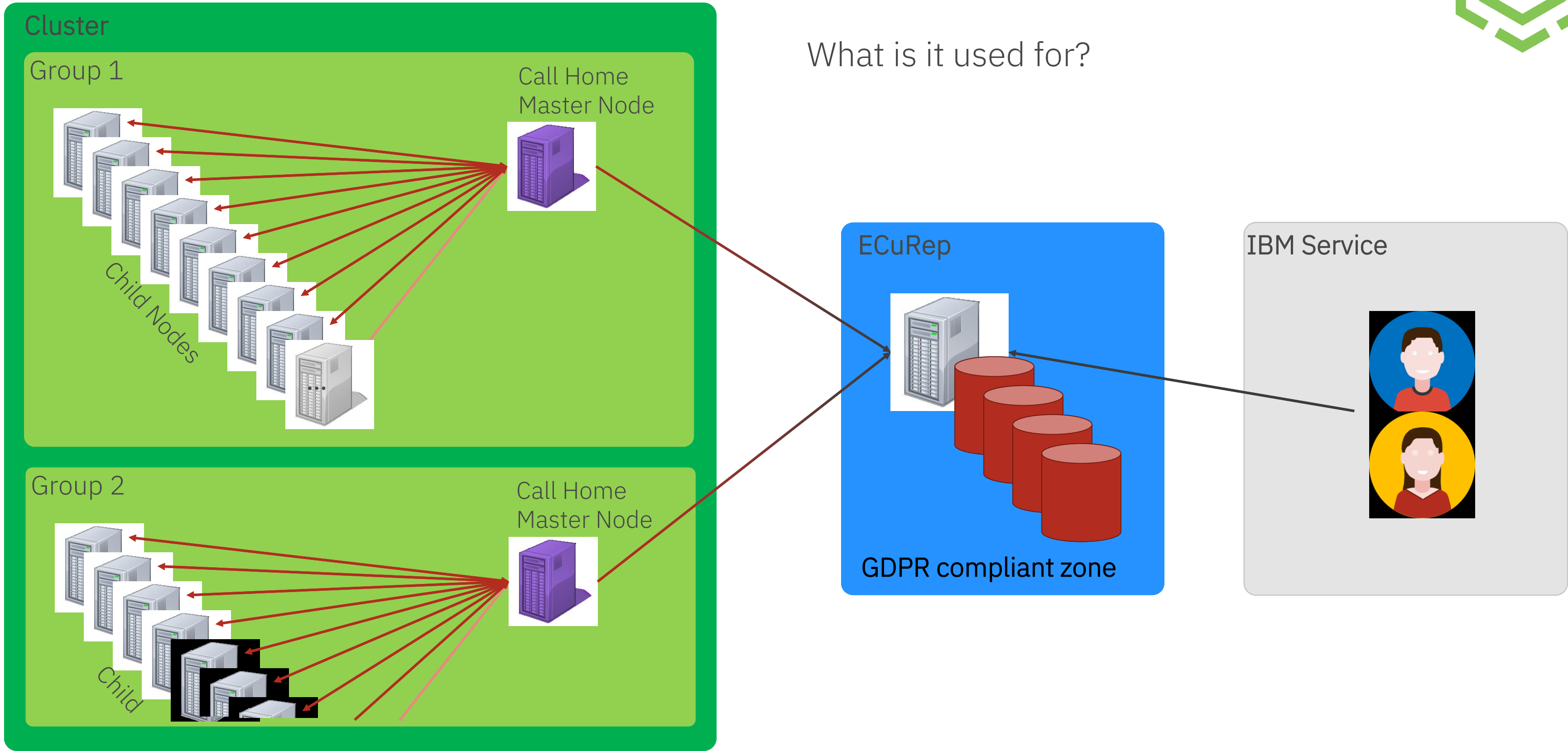
Call Home

Please enable Call Home.
Enables us to fill and run our backend infrastructure to analyse the data and to provide cognitive insights.

Helps us to:

Find issues in your cluster configuration

Provide proactive action suggestions such as configuration changes or release upgrades

Improve test coverage in our test labs based on which workloads should be tested

Understand which functionalities / components are mainly used

Helps you:

Service/Support can help faster

Prevent possible outages

Understand how product is used and where are pitfalls

Focus on features/areas to improve

Call Home
Overview
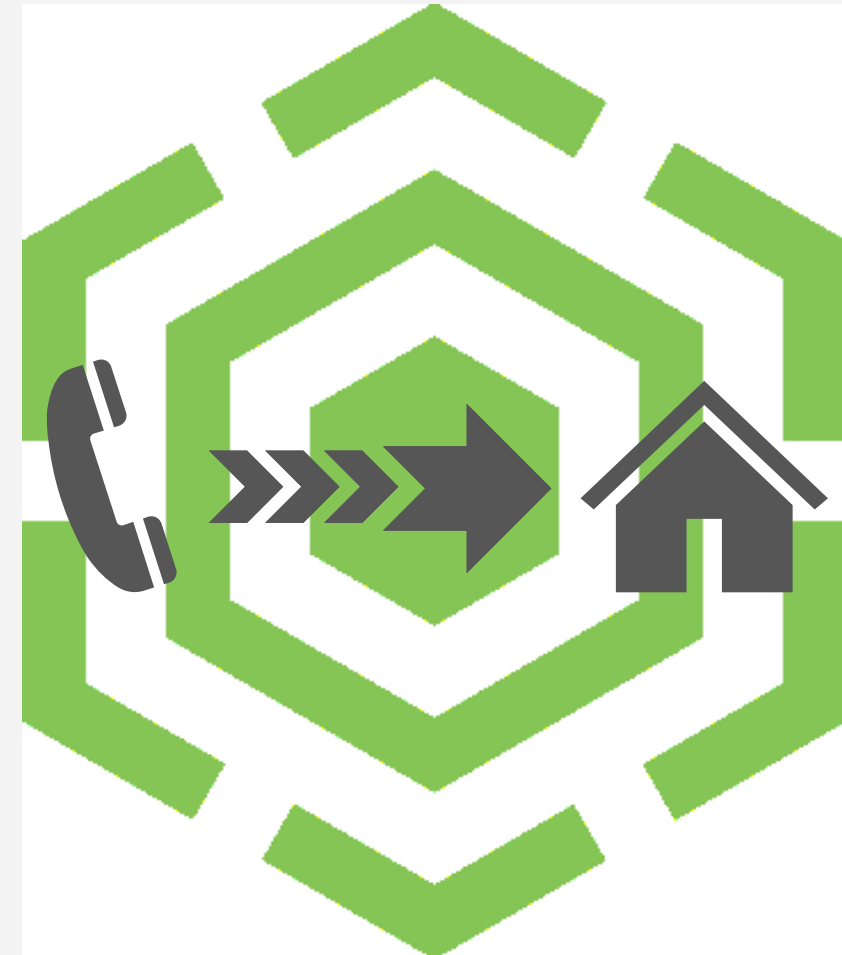
Call Home terminology

In other products this usually refers to products calling home in case of an error/incident, sending issue related information and may trigger additional actions (i.e. Service Technician to be send to exchange defect part).

SW Callhome term in context of Spectrum Scale/ESS is used as „container term", meaning:

✓ Regularly (daily/weekly) or when an event is triggered collect data and send „home".

✓ Manually send specific selected data "home" and share with IBM Service.

– Manually open a Service ticket (PMR) – Trigger to collect dumps and send „home".   Backlog

Spectrum Scale / ESS SW Call Home collects data from supported Linux nodes on all architectures (x,p,z). The difference on architecture is just the platform specifics.
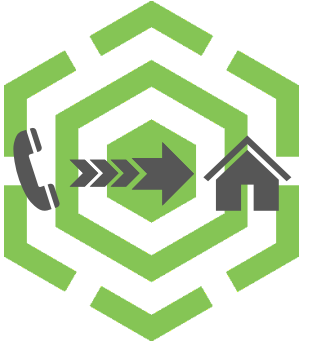
Call Home definitions

Call home group

A group of node(s) that consists of at least one node (call home node).

Call home (master) node

This node initiates the data collection within the call home group and uploads the data package to the IBM support center (ECuRep)

Call home child node
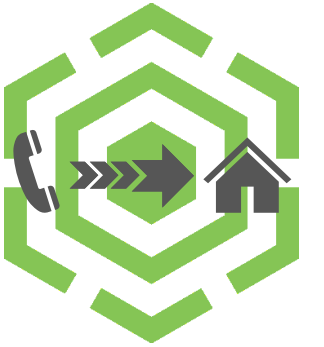
A member node of the Call home group that (locally) collects data triggered by the call home node.
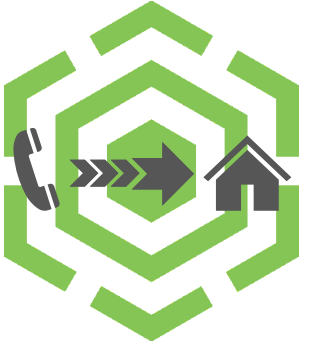
Call Home requirements

- CCR is required.

- A cluster node can only be part of ONE call home group.

- A call home group should consist at maximum (not enforced) of 32 nodes (including the call home master and child nodes per group).

- The call home master node needs to have access to (either directly or using the proxy):
  - Host name: esupport.ibm.com
  - IP address: 129.42.56.189, 129.42.60.189, and 129.42.52.189. The recommendation is to open 129.42.0.0/18.
  - Port number: 443

- To collect data from the whole cluster all nodes must be part of a call home group.

  - Multiple call home groups can be defined within an IBM Spectrum Scale cluster (max. 32 nodes per group).

- There is no fail-over capability.

  - When a call home node is unavailable the configured schedule for this call home group will fail. Manual interaction is required in case this is needed.

Recommended SW Call Home Support Matrix

Architecture	OS version	Spectrum Scale
x86_64	RHEL 7	4.2.3.7
x86_64	SLES 12 / 15	4.2.3.7 / 5.0.3.0
x86_64	UBUNTU 16.04	5.0.0.0
Power-LE	RHEL 7	4.2.3.7
Power-LE	SLES 12 / 15	4.2.3.7 / 5.0.3.0
Power-LE	UBUNTU 16.04	5.0.0.0
Power-BE	RHEL 7	4.2.3.7
Z	RHEL 7	5.0.0.0
Z	SLES 12 / 15	5.0.0.0 / 5.0.3.0
Z	UBUNTU 16.04	5.0.0.0

Best to start with
v5.0.2.x
or above
(master nodes)

All architectures (x, p, z) 64-bit.
Power-LE -> Little endian on Power
Power-BE -> Big endian on Power
Z -> S/390 - mainframe

Subject of change
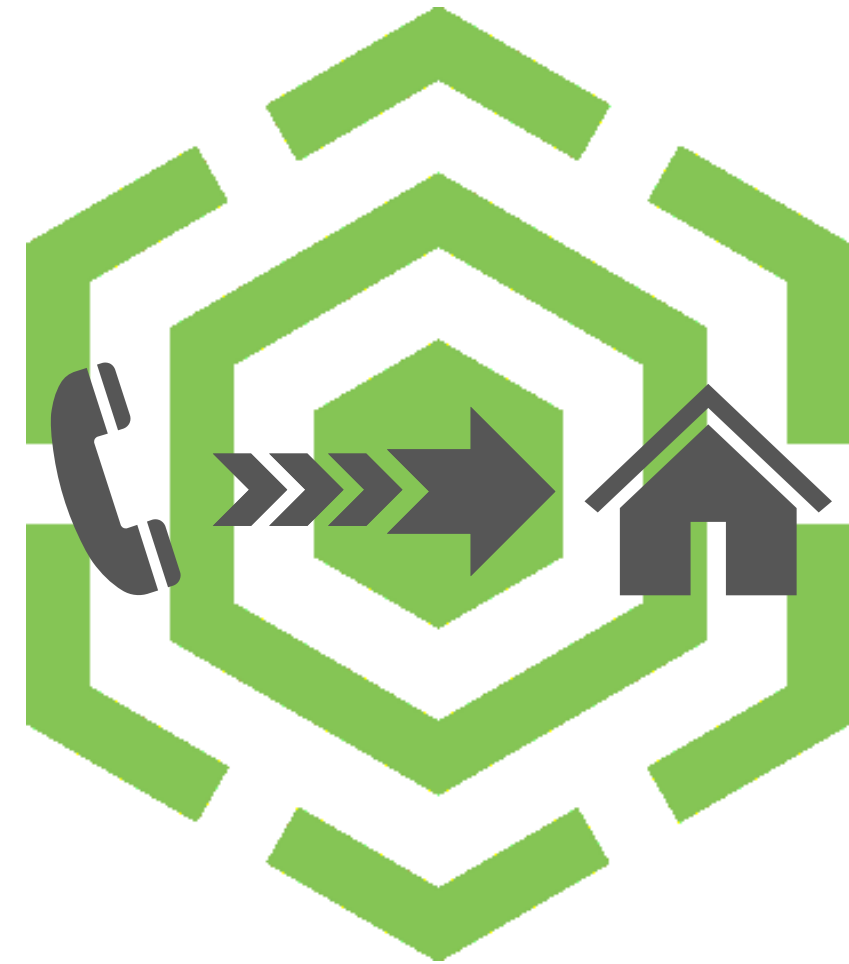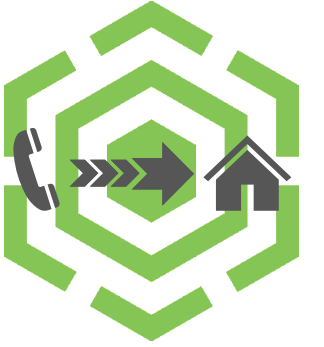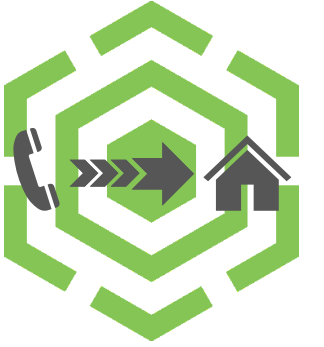
Call Home
What's new?

Call Home - What's new since 4.2.3.7 and 5.0.0?

- Lots of issues fixed (4.2.3.7 & 5.0.0)

- Install, enable and configure Call Home with the installation toolkit (5.0.0)

- Enable and configure Call Home with the GUI (5.0.0)

- Added -Y option for machine readable output (5.0.0)

- Reworked the collection list, what gets collected and how (5.0.0)

- Added --pmr option to the `mmcallhome run SendFile` command

  - allows to upload data to existing PMRs, also available in the GUI

- Added support for zLinux and Ubuntu

- Documentation updates (4.2.3.7 & 5.0.0)

Call Home – What's new (came with 5.0.2)

```
                              root@gpfs-11

Usage:
        mmcallhome group add GroupName server [--node {all | ChildNode[,ChildNode...]}]
        mmcallhome group list [--long] [-Y]
        mmcallhome group delete GroupName
        mmcallhome group auto [--server {ServerName1[,ServerName2...]}]
                              [--nodes {all | ChildNode1[,ChildNode2...]}]
                              [--force]
                              [--group-names {group1[,group2...]}]
                              [--enable [{LICENSE | ACCEPT}] | --disable]
```

If `--server` is used and n servers are specified, N groups will be created and each specified server will be a server of one group (Command changed semantically).

If `--server` is not used, all nodes will be checked and those, which have proper packages + internet connectivity, will be chosen to be potential call home group masters (servers). Of those only as many will be selected, as many are necessary to create groups with at **most** 32 nodes. However if less potential call home group masters are available, all will be taken.

In addition to node names, also node classes (e.g. linuxNodes), node IPs and GPFS node IDs can be used.

Call Home – What's new (came with 5.0.2)
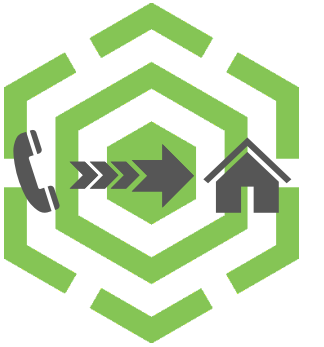
```
                                    root@gpfs-11
Usage:
        mmcallhome group add GroupName server [--node {all | ChildNode[,ChildNode...]}]
        mmcallhome group list [--long] [-Y]
        mmcallhome group delete GroupName
        mmcallhome group auto [--server {ServerName1[,ServerName2...]}]
                              [--nodes {all | ChildNode1[,ChildNode2...]}]
                              [--force]
                              [--group-names {group1[,group2...]}]
                              [--enable [{LICENSE | ACCEPT}] | --disable]
```

If `--nodes` is defined, then only the specified nodes will be distributed into groups
(helpful if you have 15000 nodes in the cluster but only want to distribute 100 nodes into call home
groups).

If `--group-names` is defined and the number of the specified group names is bigger than the number of
groups to create then only the first group names will be taken.
If the number of the specified group names is smaller than the required number of groups, then the
operation will be aborted with an error.

Call Home – What's new (comes with 5.0.3)

Call Home heartbeat

- Runs on call home master nodes.

- Sends a heartbeat once per day if capability is enabled.

- Can manually be triggered:
  `mmhealth node show callhome --refresh`

and brings new Call Home Service

- mmhealth reports if no connection to ECuRep

- Does not have effect on overall node state but indicates an issue with sending data.

<u>Data the heartbeat gathers:</u>

```
mmdiag --version -Y        /proc/meminfo

mmsysmonc d cfgshow        /proc/uptime

/bin/df -k -t gpfs         mmlslicense -Y

/bin/lsblk                 mmlsconfig -Y

/usr/bin/lscpu

/etc/os-release

/proc/cpuinfo
```

```
[root@g5030-11 ~]# mmhealth node show

Node name:       g5030-11d.localnet.com
Node status:     HEALTHY
Status Change:   5 min. ago


Component        Status        Status Change       Reasons
------------------------------------------------------------------------------------
GPFS             HEALTHY       5 min. ago          -
NETWORK          HEALTHY       5 min. ago          -
FILESYSTEM       HEALTHY       5 min. ago          -
DISK             HEALTHY       5 min. ago          -
CALLHOME         FAILED        Now                 callhome_heartbeat_failed
PERFMON          HEALTHY       5 min. ago          -
THRESHOLD        HEALTHY       5 min. ago          -
```

Call Home – What's new (comes with 5.0.3)

- By manual installation call home child nodes **only** requires gpfs.base

- File callhomeSchedules.json to verify which data is send for DAILY/WEEKLY jobs.

- Improved execution time

- New options to snap.py framework

- Support for SLES15

- Simplified package / archive structure

```
CH_20181022133123.228_daily/
        ├── CH_20181022133123.228.tar
        └── HEADER

Content of internal tar:
CH_20181022133123.228
        ├── LC_g5020-31.tar.gz
        └── LC_g5020-32.tar.gz

Together unpacked:
CH_20181022133123.228_daily/
        ├── CH_20181022133123.228
        │       ├── LC_g5020-31.tar.gz
        │       └── LC_g5020-32.tar.gz
        ├── CH_20181022133123.228.tar
        └── HEADER

                        ▼

CH_20181022133123.228_daily/
        ├── HEADER
        ├── MC_g5020-31.tar.gz
        └── LC_g5020-32.tar.gz
```
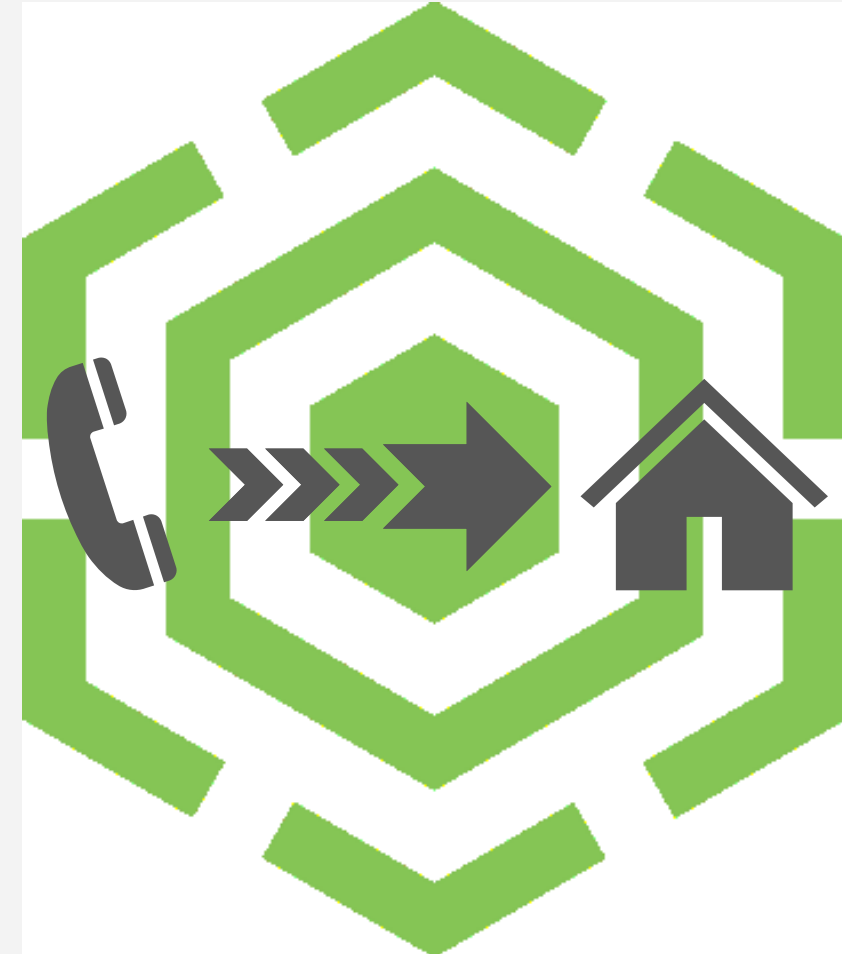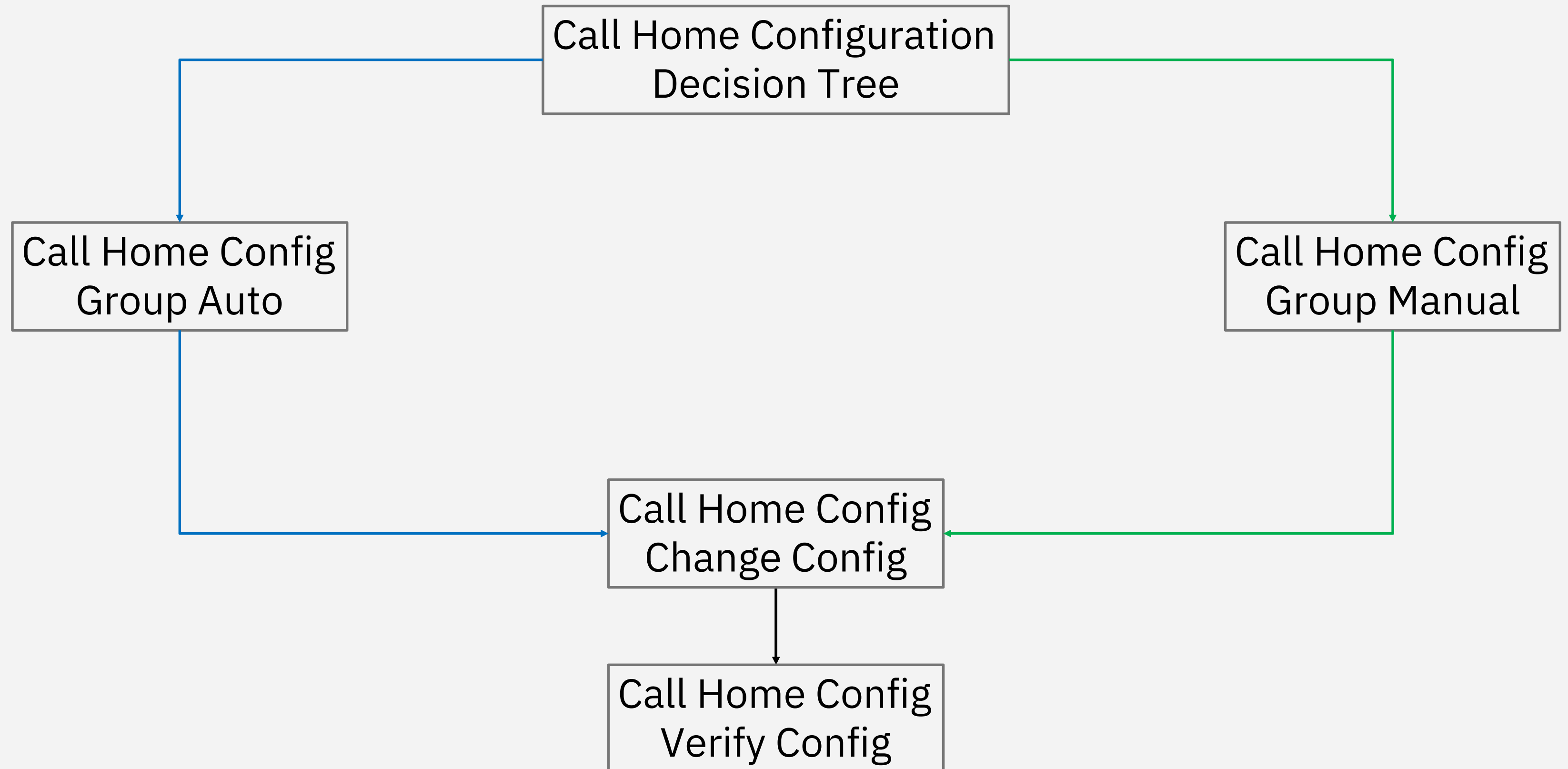
Call Home
Configuration

Call Home
Configuration Steps Overview

Call Home Configuration Decision Tree

Call Home Config Group Auto

Call Home Config Group Manual

Call Home Config Change Config

Call Home Config Verify Config

Call Home Configuration Decision Tree

- Not a <u>Dark</u> OR <u>Hippa</u> OR <u>Blue Diamond</u> Account?

- All nodes are supported call home nodes?
  (see "Recommended Support Matrix")

- All nodes have v4.2.3.7 or higher?

Then continue, else: Do NOT configure Call Home

- Not a Multi-Cluster OR AFM environment?

- Not more than 128 nodes in the cluster?

- Not a stretched cluster environment?

- Not a mixed Spectrum Scale version environment?

Then use AUTO configuration, else MANUAL
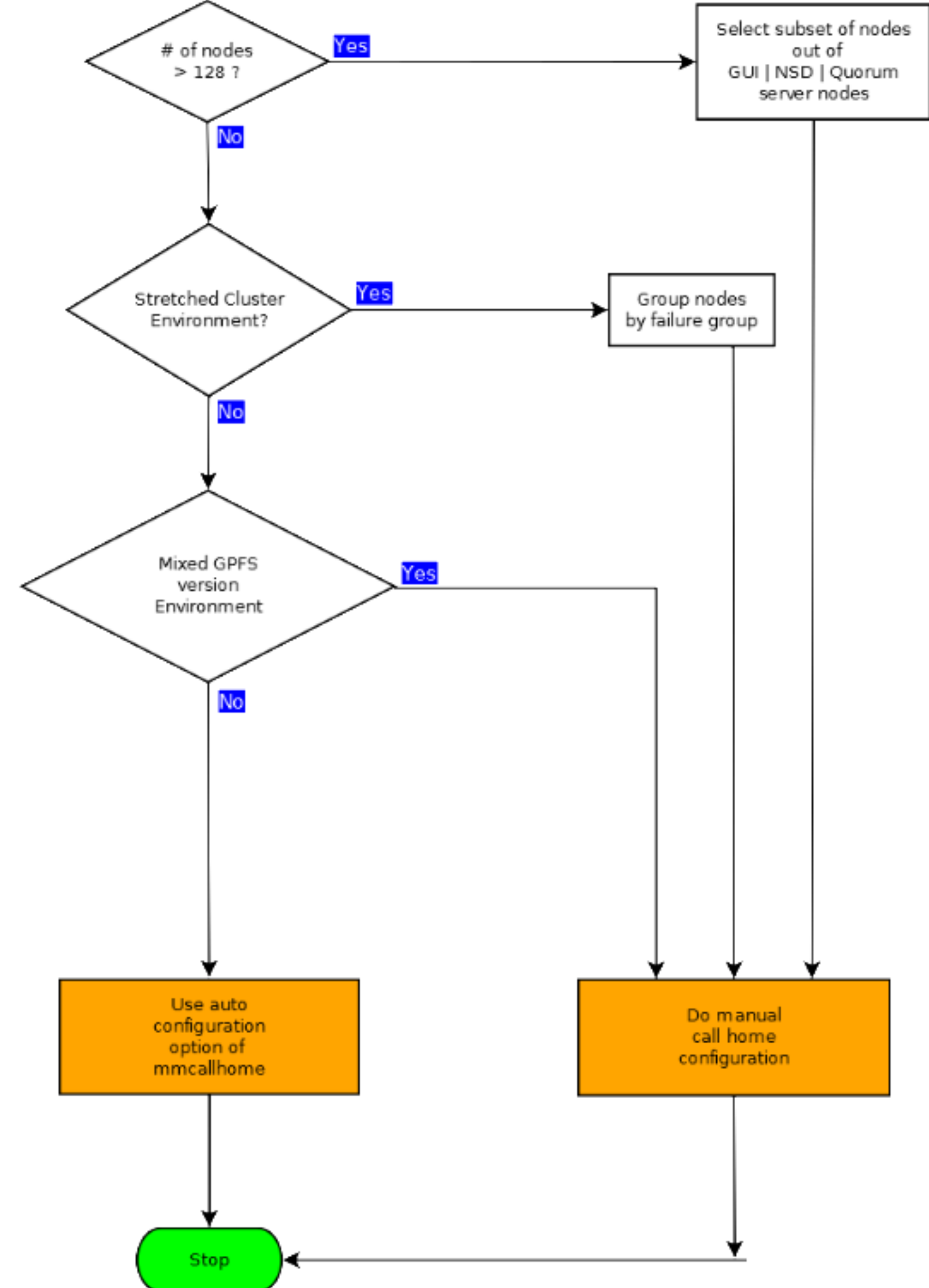
Call Home
AUTO Configuration Path

1. Commands will be executed on a supported call home node!

2. All nodes have v4.2.3.7 or higher?
   All selected nodes are neither v4.2.3.7 OR 5.0.0.0 OR 5.0.0.1 AND folders /usr and /tmp are on the same file system. Then continue, else: Do MANUAL group configuration

3. Create call home info

4. No proxy required.
   else: Configure call home proxy

5. Create call home groups

6. Enable call home

7. Verify call home configuration

```
                    root@<call home node>

mmcrnodeclass CHnodes1 -N gpfs11,gpfs12,gpfs13,gpfs14

mmcallhome info change --customer-name IBM_cl1
                      --customer-id 123456
                      --email jdoe@ibm.com
                      --country-code US

mmcallhome group auto --server gpfs-11
                      --nodes CHnodes1
                      --group-names CallhomeGroup1

mmcallhome capability enable ACCEPT

mmcallhome schedule add --task DAILY

mmcallhome schedule add --task WEEKLY


Optional:
mmcallhome proxy change --proxy-location proxy.ibm.com
                        --proxy-port 1234
                        --proxy-username user1
                        --proxy-password TopSecret
```

Subject of change

Call Home
MANUAL Configuration Path

1. Commands will be executed on a supported call home node!

2. No proxy required.
    else: Configure call home proxy

3. Syn replication not used.
    Else: Split call home nodes by replication group.

4. No other reasons to group nodes.
    Else: Split call home nodes by other reasons.

5. Number of selected nodes is NOT greater than 32.
    Else: Split groups each containing max. 32 nodes

=> Pool of intended call home groups
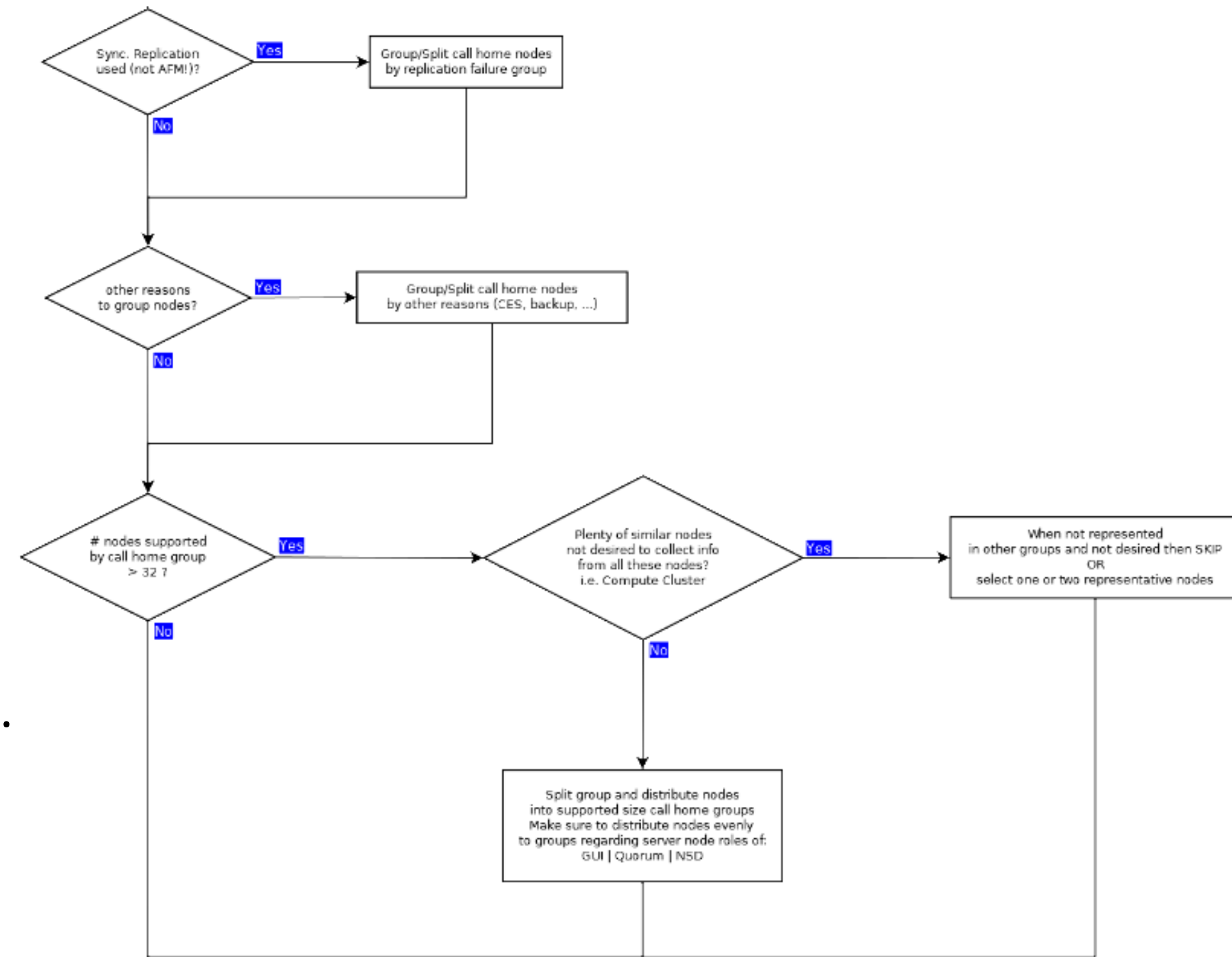
Call Home
MANUAL Configuration Path

1. Create call home info

2. No proxy required.
   else: Configure call home proxy

3. For every identified call home group:

   a) Select for each group a master call home node.

   b) Change to selected the master call home node.

   c) Test connection to IBM eSupport server.

   d) Create call home node explicitly stating master and child nodes.

4. Add scheduled tasks.

5. Enable call home.

6. Verify call home configuration.

```
                        root@<call home node>

mmcallhome info change --customer-name IBM_cl1
                        --customer-id 123456
                        --email jdoe@ibm.com
                        --country-code US

Repeat for every group:
        mmcallhome group add GroupName MASTER
                            --node child1,child2,...

mmcallhome capability enable ACCEPT

mmcallhome schedule add --task DAILY

mmcallhome schedule add --task WEEKLY


Optional:
mmcallhome proxy change --proxy-location proxy.ibm.com
                        --proxy-port 1234
                        --proxy-username user1
                        --proxy-password TopSecret
```

Call Home
VERIFY Configuration Path

Verify the configuration of:

- Call home "info" (Customer/Contact details).

- Existing Call Home group(s).

- If Call Home is enabled or not.

- Which tasks are scheduled?

- The status of executed tasks and their progress.

```
root@<call home node>

mmcallhome info list

When required:
        mmcallhome proxy list

mmcallhome group list

mmcallhome capability list

mmcallhome schedule list

mmcallhome status list
```

Call Home Configuration – mmcallhome group

Tip: Create node classes to group nodes to be used for group auto options `--server/--nodes`

- Automatically creates call home groups but provides options to influence.

- Using `--force` option removes the existing status information.

```
                                    root@gpfs-11

Usage:
        mmcallhome group add GroupName server [--node {all | ChildNode[,ChildNode...]}]      Group MANUAL
        mmcallhome group list [--long] [-Y]
        mmcallhome group delete GroupName
        mmcallhome group auto [--server {ServerName1[,ServerName2...]}]                       Group AUTO
                              [--nodes {all | ChildNode1[,ChildNode2...]}]
                              [--force]
                              [--group-names {group1[,group2...]}]
                              [--enable [{LICENSE | ACCEPT}] | --disable]
```
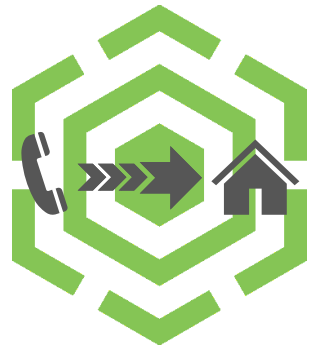
Proxy
Required?

Subject of change
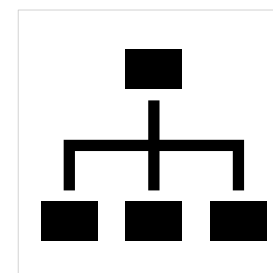
Call Home configuration via CLI or GUI

```
                              root@gpfs-11

[root@gpfs-11 installer]# ./spectrumscale callhome
usage: spectrumscale callhome [-h]
                              {enable,disable,list,schedule,clear,config} ...

[root@gpfs-11 ~]# mmcallhome -h
Usage:
        mmcallhome capability    Enable, disable and list call home capability.
        mmcallhome group         Administer topology, groups, server and client nodes.
        mmcallhome info          Administer customer data.
        mmcallhome proxy         Administer proxy configuration.
        mmcallhome run           Run a single gather send task.
        mmcallhome schedule      Administer call home scheduling.
        mmcallhome status        Shows status information.
        mmcallhome test          Run a system check.
```

Call Home

☐ Enable Call Home

Call home node:

gpfs-11.localnet.com

Company Information

Company name: Customer ID:

E-mail: Country:
                                       -

Proxy Information

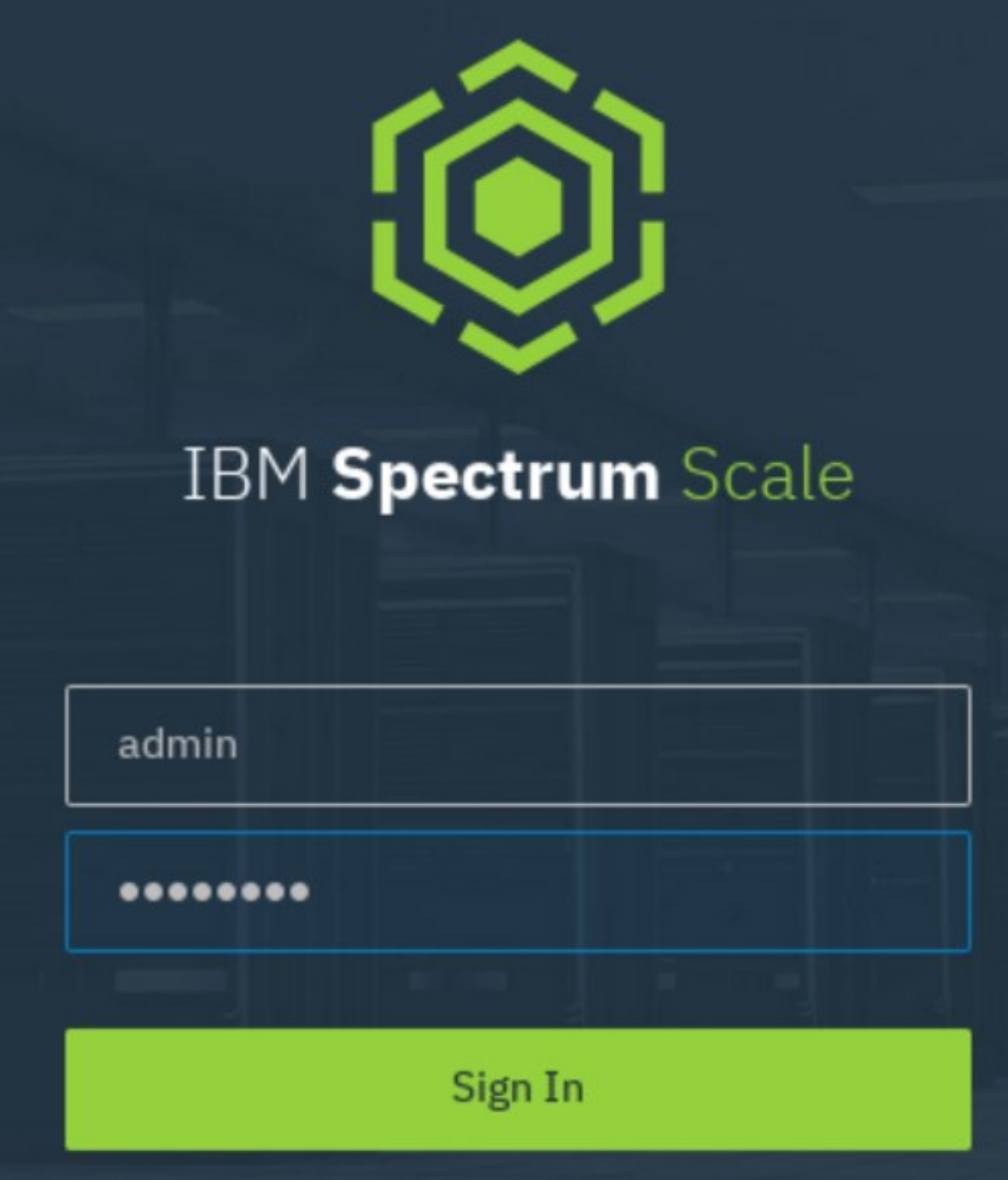☐ Enable Proxy

Proxy host: Proxy port:
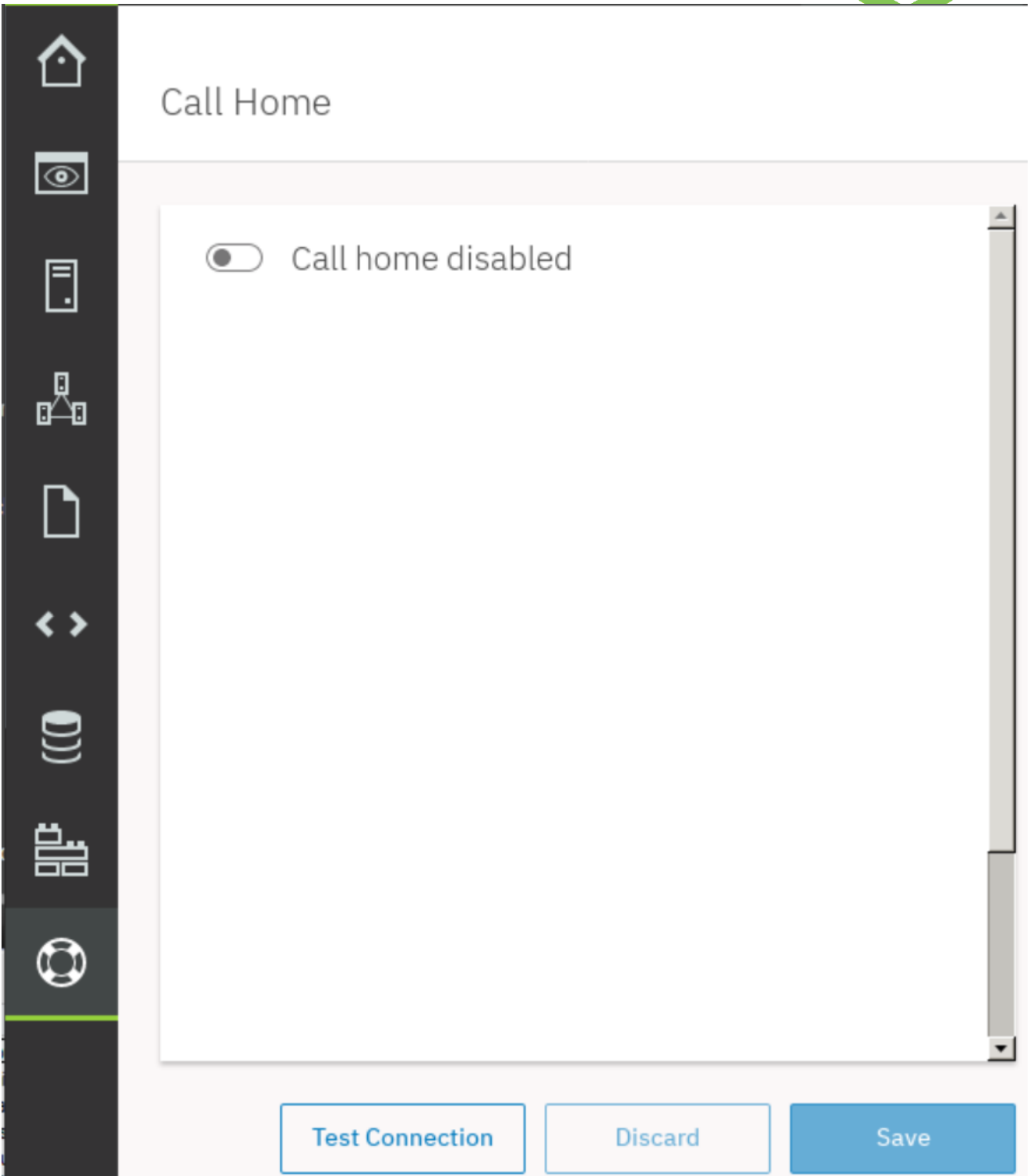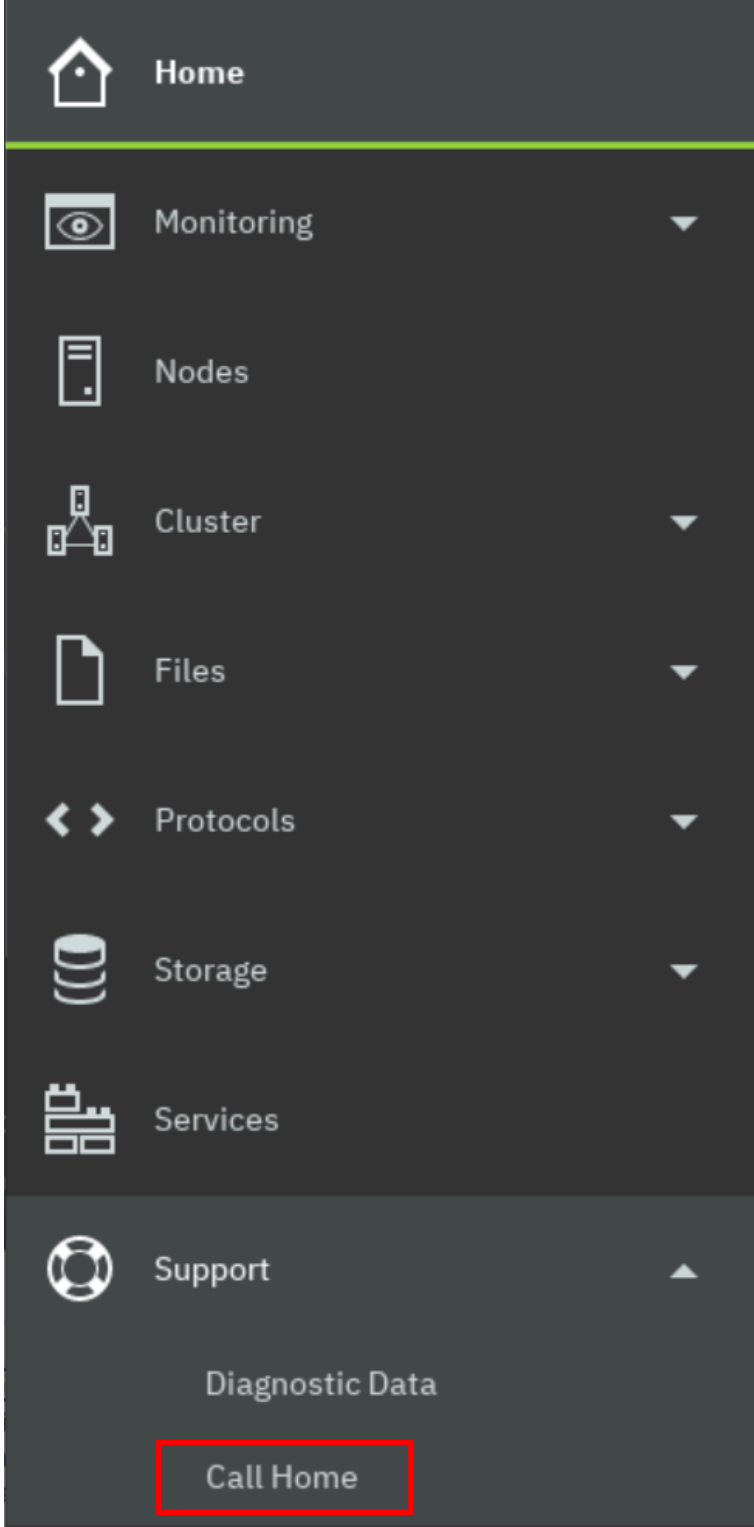
☐ Enable Proxy authentication

Proxy username: Proxy password:

Test Connection   Configure Call Home

https://<gui node IP/name>/gui#support-callhome

Call Home GUI panel
Configure Call Home

Call Home GUI panels
  Call Home Configuration

GUI supports only one Call Home Group!
Use CLI instead in case more groups are required.

```
                                                    root@gpfs-11

# mmcallhome info list
group        customerName      customerID      email              countryCode
--------     --------------    ------------    --------------     -----------
global       IBM_TEST          123456          jdoe@ibm.com       US

# mmcallhome group list
callHomeGroup      callHomeNode      callHomeChildNodes
--------------     -------------     -----------------------------
-
autoGroup_1        gpfs-11          gpfs-11,gpfs-12,gpfs-13,gpfs-14
```

Call Home GUI panel
Call Home Upload Diagnostic Data

Call Home mmcallhome group auto

Tip:
Create node classes to group nodes to be used for group auto options --server/--nodes
i.e. for CES/GUI nodes, groups per Stretched/Multi-Cluster site, ...
`mmcrnodeclass ClassName –N {Node[,Node...] | NodeFile | NodeClass}`

```
                                    root@gpfs-11

mmcrnodeclass CallhomeCandidates -N gpfs-11,gpfs-12,gpfs-13,gpfs-14

mmcallhome info change --customer-name IBM_TEST --customer-id 123456 --email jdoe@ibm.com --country-code US
[mmcallhome proxy change …]
mmcallhome group auto --server gpfs-11 --nodes CallhomeCandidates –group-names CallhomeGroup1 –enable ACCEPT
[mmcallhome capability enable ACCEPT]
mmcallhome schedule add --task DAILY
mmcallhome schedule add --task WEEKLY
```

Text example for: `mmcallhome group auto` with options

There are 100 nodes to distribute and 10 nodes have internet connectivity
=> 4 groups will be created with 25 nodes in each

There are 100 nodes to distribute and 2 nodes have internet connectivity
=> 2 groups will be created with 50 nodes in each

HINT: In addition to node names, also node classes (e.g. linuxNodes), node IPs and GPFS node IDs with all relevant options of mmcallhome can be used.

Subject of change

Call Home Configuration - Simple Example - Verify

Verify the configuration (New node classes created (in blue) by mmcallhome command)

```
                                    root@gpfs-11

# mmlsnodeclass
Node Class Name        Members
---------------------- ------------------------------------------------------------
CallhomeCandidates     gpfs-13.novalocal,gpfs-11.novalocal,gpfs-12.novalocal,gpfs-14.novalocal
CALLHOME_SERVERS       gpfs-11.novalocal

# mmcallhome info list
group     customerName                       customerID    email               countryCode
--------  ---------------------------------  ------------  ------------------  -------------
global    IBM_TEST                           123456        jdoe@de.ibm.com     US

# mmcallhome group list
callHomeGroup    callHomeNode    callHomeChildNodes
---------------- -------------- -----------------------------------
CallhomeGroup1   gpfs-11        gpfs-11,gpfs-12,gpfs-13,gpfs-14

# mmcallhome capability list
group     callHomeNode    status
--------  -------------- ---------
global    ---             enabled
```
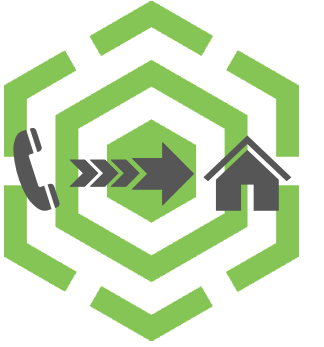
Call Home Configuration – Stretched Cluster Example

Cluster with 40 nodes with sync replication between site A & B plus tiebreaker node in site C.

- Node 1 is GUI node (in site A&B - pick nodes as call home server)

- Node 2-4 are NSD server nodes (in site A&B – pick nodes as call home child nodes).

- Rest of nodes (5-20) are computing nodes – just pick one of them as representatives

```
                                         root@gpfs-11

[mmcrnodeclass ChServerCandidatesA -N gpfs-A1]
mmcrnodeclass ChChildCandidatesA -N gpfs-A2,gpfs-A3,gpfs-A4,gpfs-A9
[mmcrnodeclass ChServerCandidatesB -N gpfs-B1]
mmcrnodeclass ChChildCandidatesB -N gpfs-B2,gpfs-B3,gpfs-B4, gpfs-B15
[mmcrnodeclass ChServerCandidatesC -N gpfs-C1]

mmcallhome info change --customer-name IBM_cl1 --customer-id 123456 --email jdoe@ibm.com --country-code US
mmcallhome group auto -server gpfs-A1 --nodes ChChildCandidatesA –-group-names CallhomeSiteAGroup1
mmcallhome group auto -server gpfs-B1 --nodes ChChildCandidatesB –-group-names CallhomeSiteBGroup1
mmcallhome group auto -server gpfs-C1 --group-names CallhomeSiteCGroup1
mmcallhome capability enable ACCEPT
mmcallhome schedule add --task DAILY
mmcallhome schedule add --task WEEKLY
```

Subject of change
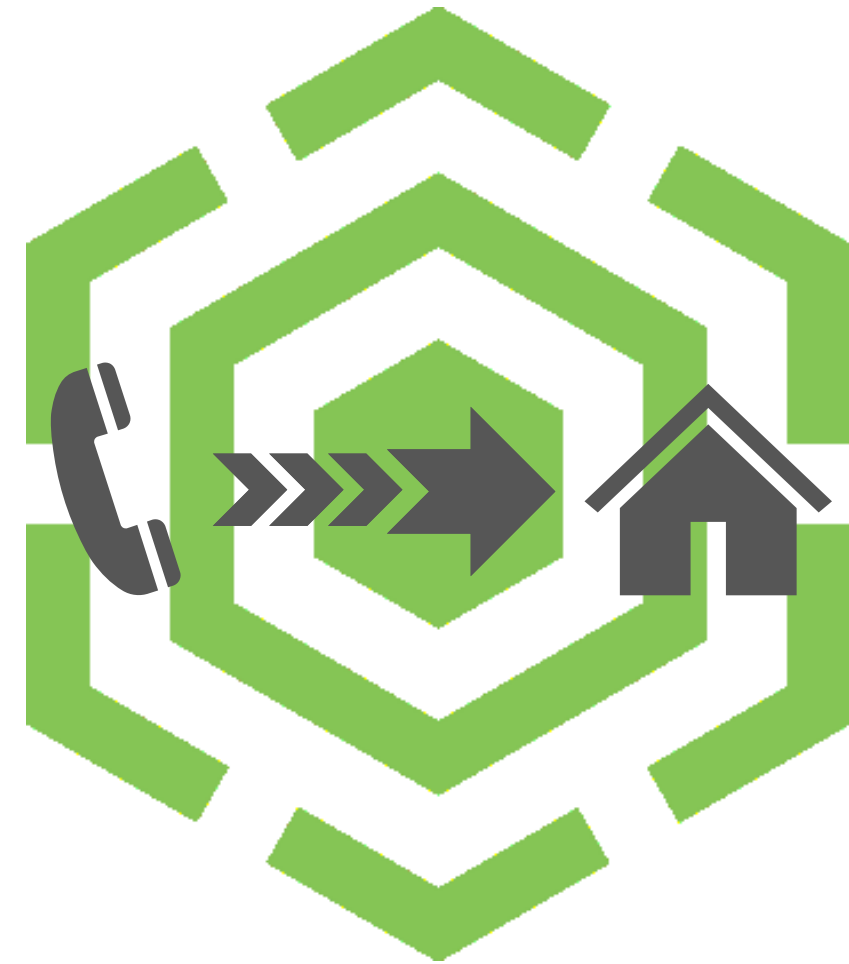
Call Home Usage

Call Home Usage – mmcallhome info

Command to configure Customer/Contact details required to upload data to IBM eServer

```
root@<call home node>
Usage:
        mmcallhome info list
        mmcallhome info change
        mmcallhome info change --customer-name CustomerName
        mmcallhome info change --customer-id CustomerId
        mmcallhome info change --email Email
        mmcallhome info change --country-code CountryCode
```
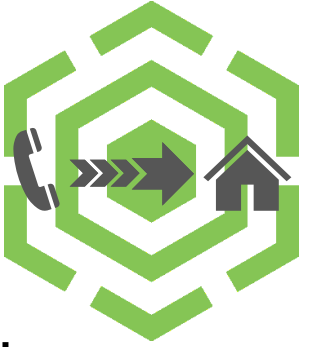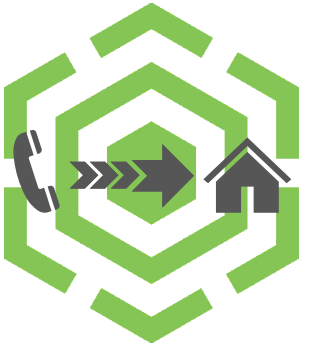
Call Home Usage – mmcallhome proxy

Command to configure a proxy server in case nodes cannot access the required IBM eServer directly

```
Usage:
        mmcallhome proxy enable [--with-proxy-auth]
        mmcallhome proxy disable
        mmcallhome proxy list
        mmcallhome proxy change
        mmcallhome proxy change --proxy-location ProxyLocation
        mmcallhome proxy change --proxy-port ProxyPort
        mmcallhome proxy change --proxy-username ProxyUsername
        mmcallhome proxy change --proxy-password ProxyPassword
```

Call Home Usage – mmcallhome group

Tip: Create node classes to group nodes to be used for group auto options `--server/--nodes`

- Automatically creates call home groups but provides options to influence.

- Using `--force` option for group auto removes the existing call home status information!

```
                                    root@gpfs-11

Usage:
        mmcallhome group add GroupName server [--node {all | ChildNode[,ChildNode...]}]
        mmcallhome group list [--long] [-Y]
        mmcallhome group delete GroupName
        mmcallhome group auto [--server {ServerName1[,ServerName2...]}]
                              [--nodes {all | ChildNode1[,ChildNode2...]}]
                              [--force]
                              [--group-names {group1[,group2...]}]
                              [--enable [{LICENSE | ACCEPT}] | --disable]
```

Group MANUAL

Group AUTO

Call Home Usage – mmcallhome capability
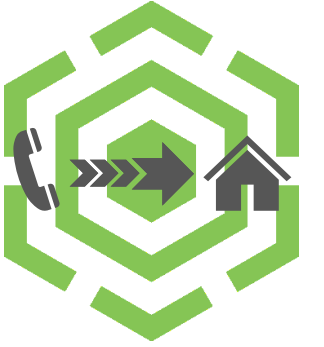
Command to enable/disable call home.

root@<call home node>

Usage:
      mmcallhome capability list
      mmcallhome capability enable [{LICENSE | ACCEPT}]
      mmcallhome capability disable

Call Home Usage – mmcallhome run GatherSend

On a node of a call home group.

Submits cluster/system information triggered manually or on a scheduled base.

Be patient after starting the command before it returns control back to console.
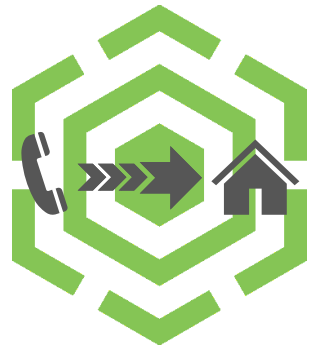Use `mmcallhome status list` (in other console) to see progress.

```
                                    root@gpfs-11

Usage:
        mmcallhome run GatherSend --task {DAILY | WEEKLY}

# mmcallhome run GatherSend --task DAILY
One time run completed with success

# mmcallhome status list -v --numbers 1 --task DAILY
=== Executed call home tasks ===

Group          Task    Start Time           Updated Time     Status    RC or Step  Package File Name
                       Original Filename
-----------------------------------------------------------------------------------------------autoGroup_1
daily  20190225094721.281  20190225095704   success   RC=0
/tmp/mmfs/callhome/rsENUploaded/14457945700502.5_0_3_0.123456.US.TEST.autoGroup_1.gat_daily.g_daily.scale.20190225094721281
.cl0.DC
```

Call Home Usage – mmcallhome run SendFile
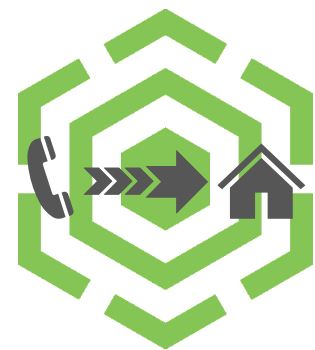
On a node of the selected call home group.

Submits the specified file(s).

```
Usage:
        mmcallhome run SendFile --file file [--desc DESC | --pmr {xxxxx.yyy.zzz | TSxxxxxxxxx}]

# mmcallhome run SendFile --file /tmp/err.log --desc "Test file ..."
Running sendFile... (In case of network errors, it may take over 20 minutes for retries.)
Successfully uploaded the given file
Run mmcallhome status list --verbose to see the package name

# mmcallhome status list --task SENDFILE -v --numbers 1
=== Executed call home tasks ===

Group           Task        Start Time              Updated Time    Status    RC or Step       Package File Name
                                  Original Filename
-------------------------------------------------------------------------------------------------autoGroup_1
sendfile  20190225161211.761  20190225161211  running  Uploading package
/tmp/mmfs/callhome/rsENHomeQ/14457945700502.5_0_3_0.123456.US.TEST.autoGroup_1.Test_file____.s_file.scale.20190225161211761
.cl0.DC  /tmp/err.log
```

Call Home GUI panel
Call Home Upload Diagnostic Data

Call Home Usage – mmcallhome status list

Display status/progress of tasks.

Option `--numbers` is per item {DAILY | WEEKLY | SENDFILE | SENDPMRDATA}

```
                                        root@gpfs-11

Usage:
        mmcallhome status list [--task {DAILY | WEEKLY | SENDFILE | SENDPMRDATA}]
                               [--numbers num] [--verbose] [-Y]

# mmcallhome status list                        # mmcallhome status list --numbers 2
=== Executed call home tasks ===                === Executed call home tasks ===

Group         Task      Start Time        Status   Group         Task      Start Time        Status
-------------------------------------------------  -------------------------------------------------
autoGroup_1   daily     20190222020401.521  success  autoGroup_1   daily     20190225094721.281  success
autoGroup_1   daily     20190223020401.906  success  autoGroup_1   daily     20190226020401.755  success
autoGroup_1   daily     20190224020401.406  success  autoGroup_1   weekly    20190224031501.651  success
autoGroup_1   daily     20190225020401.669  success  autoGroup_1   sendfile  20190225161211.761  success
autoGroup_1   daily     20190225094721.281  success  autoGroup_1   sendfile  20190226160225.689  success
autoGroup_1   daily     20190226020401.755  success
autoGroup_1   weekly    20190224031501.651  success
autoGroup_1   sendfile  20190225161211.761  success
autoGroup_1   sendfile  20190226160225.689  running
```
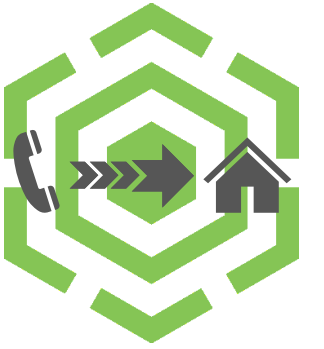
Call Home Usage – mmcallhome schedule

List/Add/Delete schedules of daily/weekly tasks.

```
Usage:
        mmcallhome schedule list [-Y]
        mmcallhome schedule add --task {DAILY | WEEKLY}
        mmcallhome schedule delete --task {DAILY | WEEKLY}

# mmcallhome schedule list
=== List of registered schedule tasks ===

group     scheduleType     isEnabled     confFile
--------  --------------   -----------   ----------
global    daily            enabled       daily.conf
global    weekly           enabled       weekly.conf
```

Call Home Usage – mmcallhome test connection

Verify the call home (master) node can reach out to the IBM eSupport Server.

```
# mmcallhome test connection
## Starting connectivity test between the call home node and IBM
Call home node: gpfs-11.novalocal
Starting time: Tue Feb 26 15:52:47 2019

Testing direct connection
-------------------------------------------------------------
Testing <prefix Edge_SP_Config>:
Edge_SP_Config_1: 129.42.56.189 OK

Testing <prefix Edge_Profile>:
Edge_Profile_1: 129.42.56.189 OK

Testing <prefix Edge_Status_Report>:
Edge_Status_Report_1: 129.42.56.189 OK


-------------------------------------------------------------
End time: Tue Feb 26 15:53:03 2019
```

Call Home
Data Collection
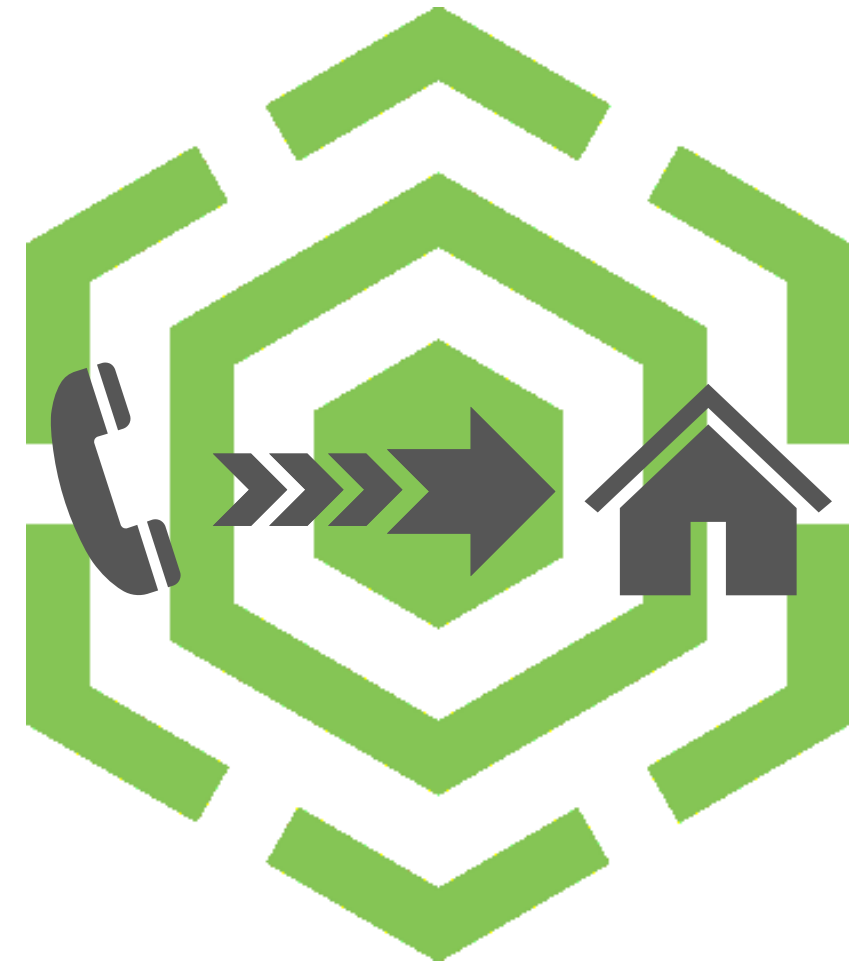
What does Call Home GatherSend (daily/weekly) collect?
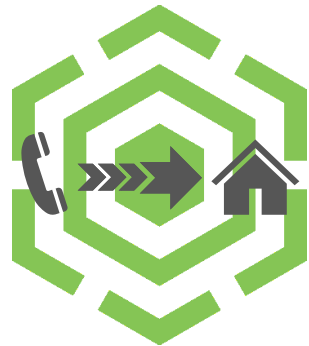
Documented in [Knowledge Center](#) (change to your Spectrum Scale version).

command	machine-type	node	os	product	schedule
lscpu	all	all	all	all	all
cat /proc/interrupts	all	all	all	all	all
lsblk	all	all	all	all	weekly
lsdasd -a -u	s390x	all	all	all	all
lszfcp -H -P -D	s390x	all	all	all	all
lsdasd -b -l	s390x	all	all	all	all
lsscsi -li	all	all	all	all	all
multipath -ll	all	all	all	all	all
mmdiag --config -Y	all	all	all	all	weekly
mmdiag --version -Y	all	all	all	all	all
mmlslicense -L -Y	all	CALLHOME_SERVERS	all	all	weekly
mmlsnodeclass --system -Y	all	CALLHOME_SERVERS	all	all	all
mmlscluster -Y	all	CALLHOME_SERVERS	all	all	all
mmlscluster --ces -Y	all	CALLHOME_SERVERS	all	all	all
mmces address list -Y	all	CALLHOME_SERVERS	all	all	all

Microsoft Excel
Worksheet

What does Call Home collect specifically on Systems Z?

command	machine-type	node	os	product	schedule
lsdasd -a -u	s390x	all	all	all	all
lszfcp -H -P -D	s390x	all	all	all	all
lsdasd -b -l	s390x	all	all	all	all
lsqeth -p	s390x	all	all	all	all
lscss --vpm --avail	s390x	all	all	all	all
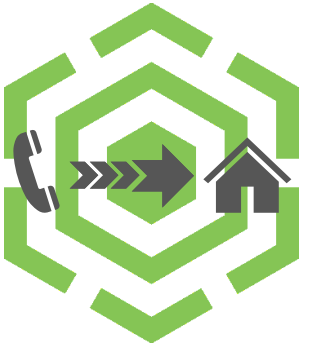
What does Call Home collect specifically on ESS?

command	machine-type	node	os	product	schedule
mmlsfirmware -Y	all	CALLHOME_SERVERS	all	ess	weekly
mmlsvdisk -Y	all	CALLHOME_SERVERS	all	ess	all
mmlscomp -Y	all	CALLHOME_SERVERS	all	ess	weekly
mmlsenclosure all -Y -N GUI_RG_SERVERS	all	CALLHOME_SERVERS	all	ess	all
mmlspdisk all -Y	all	CALLHOME_SERVERS	all	ess	all
tslsenclslot -adY	all	CALLHOME_SERVERS	all	ess	all
mmlsrecoverygroup -Y	all	CALLHOME_SERVERS	all	ess	all
mmlsrecoverygroupevents <rg> --days 2	all	CALLHOME_SERVERS	all	ess	daily
mmlsrecoverygroupevents <rg> --days 8	all	CALLHOME_SERVERS	all	ess	weekly
mmfsadm dump pdisk	all	all	all	ess	weekly
/usr/lpp/mmfs/samples/vdisk/gnrhealthcheck	all	CALLHOME_SERVERS	all	ess	weekly

Thank you

Andreas Lüngen
IT Specialist, Spectrum Scale – Proactive Services
—
luengen@de.ibm.com
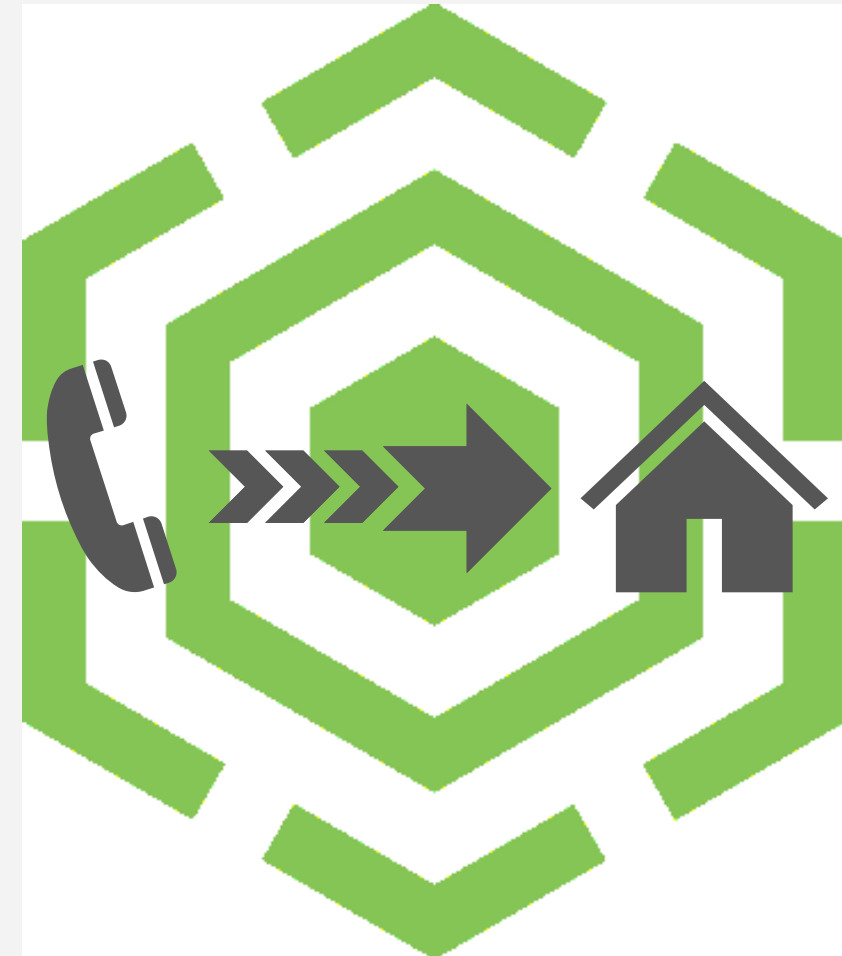
Call Home Cheat Sheet

A Call Home group consists of master and child nodes.

CCR is required.

A cluster node can only be part of ONE call home group.

Master node needs access to IBM eSupport Server.

All nodes must be part of a call home group.

There is no fail-over capability (master node)

Best to start with master nodes on v5.0.2.x or above

Cluster
Group 1   Master
Childs
...
Group 2
Master
eSupport
GDPR zone
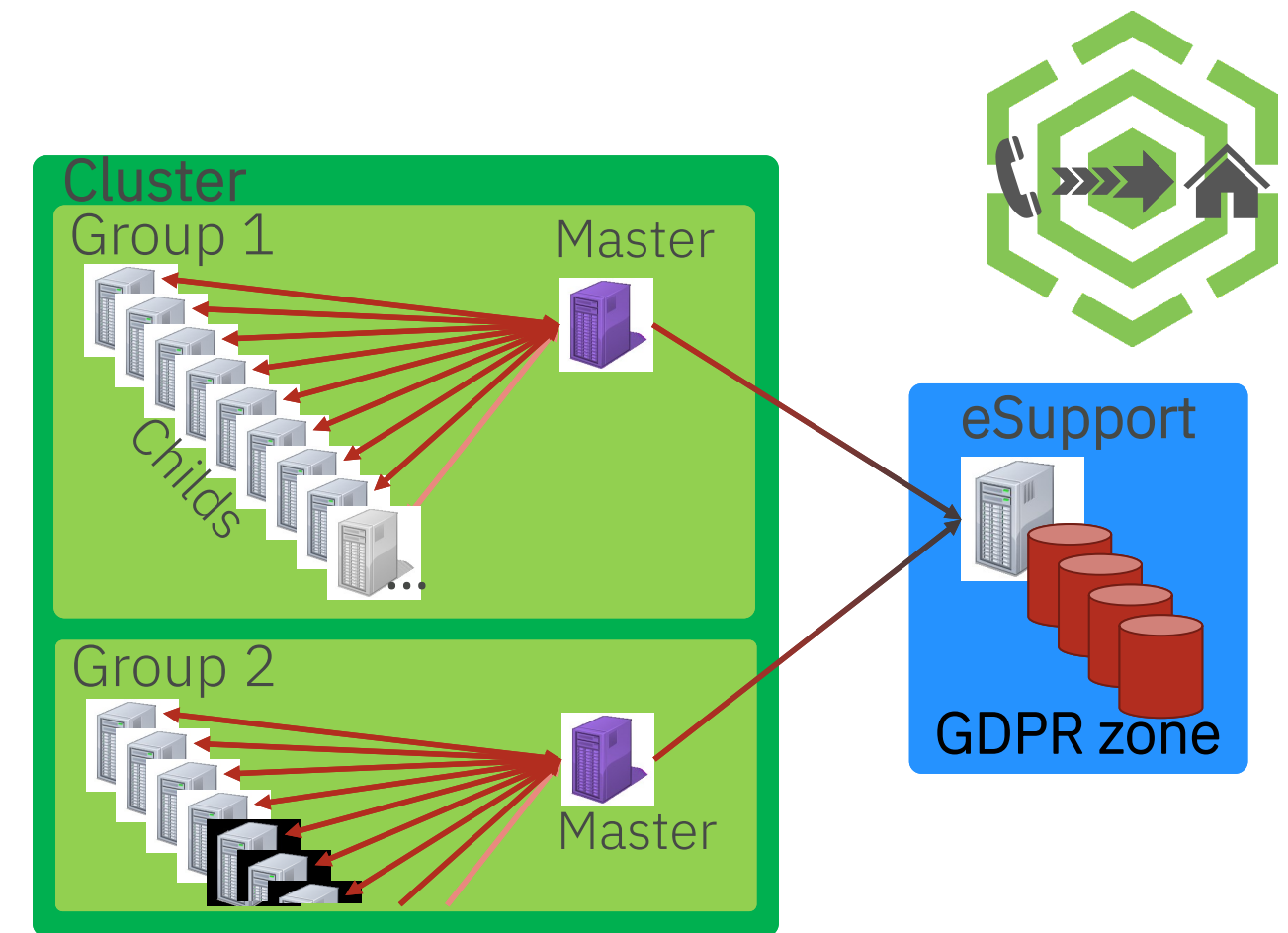
Which data gets gathered - see KC

```
root@gpfs-11

mmcrnodeclass CallhomeCandidates -N gpfs-11,gpfs-12,gpfs-13,gpfs-14

mmcallhome info change --customer-name IBM_cl1 --customer-id 123456 --email jdoe@ibm.com --country-code US
[mmcallhome proxy change …]
mmcallhome group auto --server gpfs-11 --nodes CallhomeCandidates —group-names CallhomeGroup1
mmcallhome capability enable ACCEPT
mmcallhome schedule add --task DAILY
mmcallhome schedule add --task WEEKLY
```

Call Home mmcallhome group – Proxy required?

Flowchart:

Proxy required?
- **No** → mmcallhome group auto [--server {ServerName[,ServerName...] [--nodes {all | ChildNode1[,ChildNode2...]}]}]
- **Yes** → Configure Proxy
  - → mmcallhome proxy change --proxy-location ProxyLocation
    --proxy-port ProxyPort
    [--proxy-username ProxyUsername |
    --proxy-password ProxyPassword]
  - → mmcallhome proxy enable [--with-proxy-auth]
    → (back to mmcallhome group auto)