

# ServiceNow® Relationships

Optimizing your CMDB for  
Accuracy and Visualization

Created By:

**Clemens Pfeiffer**, Tier44 Technologies, Inc.  
2394 Walsh Avenue, Santa Clara, CA 95051

**Joel Kosmich**, CDI LLC  
696 Route 46 West, Teterboro, NJ 07608

Version: v-3.1



COMPUTER  
DESIGN &  
INTEGRATION LLC



# Contents

|           |   |
|-----------|---|
| <b>03</b> | Introduction  |
| <b>04</b> | Default Relationships and Assumptions                       |
| <b>06</b> | Relationships the Right Way                                 |
| <b>06</b> | The Location Tree   |
| <b>09</b> | The Power Tree  |
| <b>11</b> | The Power and Location Tree Starting Points                 |
| <b>15</b> | Additional Tree Structures of Interest                      |
| <b>15</b> | Management Relationships                                    |
| <b>16</b> | Owner Relationships   |
| <b>18</b> | Other Relationships   |
| <b>19</b> | About CDI LLC   |
| <b>19</b> | About Tier44 Technologies, Inc.                             |
| <b>20</b> | Appendix A: Default ServiceNow CMDB Relationship Types      |
| <b>21</b> | Contact CDI LLC or Tier44 Technologies, Inc. to get started |

# Introduction

ServiceNow provides a hosted service management environment for a variety of business cases within organizations and a large set of predefined services including change management, configuration management (CMDB), workflow, and incident management. All of these have pre-defined data structures that can be extended and modified as needed. This whitepaper discusses an optimized viewpoint of defined relationship data structures and their importance specifically related to the CMDB structure. After reading the paper if you would like more information please contact CDI or Tier44 Technologies at the contact information provided at the end of the paper.

If you want to use ServiceNow to manage data center operating services, there are a variety of pre-defined templates available and some core features that can be customized to support your own operating environment. However, there is optimized relationship structures described in this paper that you should seriously consider that will result in improved accuracy, easier management, and enhanced benefits realized for your Service Management organization.

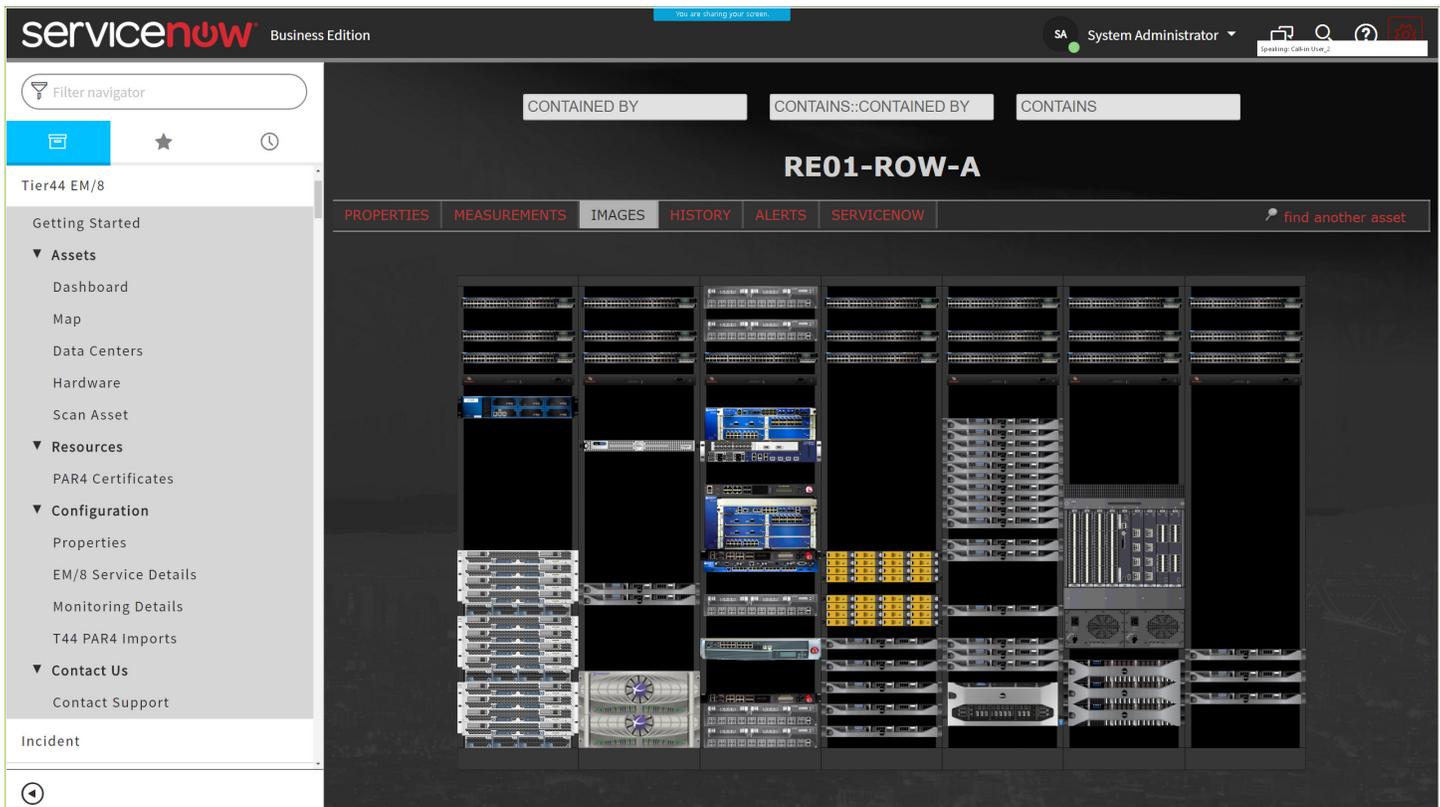


Figure 1: ServiceNow with Tier44 EM/8 Sample View

The optimized CMDB structure that we are recommending in this whitepaper provides significant benefit in the areas of service mapping, system and application incident triage, reduction in MTTR (mean time to resolution), and enablement of streamlined and more accurate audits. This optimized structure coupled with real time rack elevation views, floor plans, and monitoring can be now realized through the Tier44 EM/8® application. There is a 30-day free trial version available that you can download and install on any Geneva or Helsinki instance. Figure 1 above provides an example of a ServiceNow CMDB visualization with EM/8.

One of the most powerful and useful features in ServiceNow when it comes to data center CMDB configurations is the use of “relationships”. Relationships allow linkages of CMDB Configuration Items (“CI”) with each other. ServiceNow provides a standard set of relationships and allows you to create your own. Relationships are parent/child linkages and come in many types. Types are defined in the CMDB\_rel\_type table. Relationships can also have standard parameters and even support custom extensions. A standard relationship parameter is “port”, used as the primary parameter on a relationship. Ports are integer values similar to a rack u-position or outlet of a PDU.

## Default Relationships and Assumptions

ServiceNow provides a set of default relationships of various types and purposes. The main ones of interest for data center owners and operators are of type “contains,” “powers,” “manages” and “uses.” See Appendix A: Default ServiceNow CMDB Relationship Types for a complete list of default relationships in Helsinki.

While relationships are extremely useful, there is also a risk of defining too many similar relationships. For example, “contains,” there are a variety of “contains” relationships like “contains room,” “houses,” “zone contains,” etc. each of which has their own relationship id. The result of using them is a type-specific point-to-point relationship for a single hierarchy level.

Why is this an issue?

The main issue is a loss of the end-to-end tree structure. When using point-to-point relationship types there is no integrated navigation across the full tree based on a single type. As a result, you cannot collapse a tree and easily bridge levels. You have to jump from one relationship type to another as you traverse from top to bottom.

The second issue is that some default relationships have a parent/child relationship that is defined in a non-natural way, meaning they are not defined in a way a data center operator or IT Staff would look at parent/child relationships.

Appendix A: Default ServiceNow CMDB Relationship Types are examples of relationships with natural parent/child structures and some that you could consider backwards. An example of a natural one is “powers” and a child descriptor “powered by.” You use it by reading or interpreting it as <parent CMDB ci item> “parent descriptor” <child CMDB ci item> “child descriptor.” For example, a PDU and a server, assuming the PDU powers the server, the sentence will read “pdu powers server” and “server powered by pdu” which is the way data center operators would commonly reference the relationship.

An example of a standard definition is “In rack” and “rack contains.” Using that particular default relationship (and the assumption that a rack contains servers) you end up with “rack R1 in rack server ABC” and “server ABC rack contains rack R1,” which obviously does not make much sense. However, if you would swap the relationship it would read “server ABC in rack::rack R1” and “rack R1::rack contains server ABC.” While the swapped relationship would read fine, it is not the way the parent/ child tree should look because the server is the parent of the rack.

The dependency tree visualizes the issue:

As you can see, the server is at the top and the computer room (or even data center) is at the bottom. As a data center owner and operator you would want this to flow the other way with the data center being the parent all other connections listed underneath.

If you go through the default relationships with that in mind you find some that fit well, some that should be swapped, and some that break the tree and should not be used.

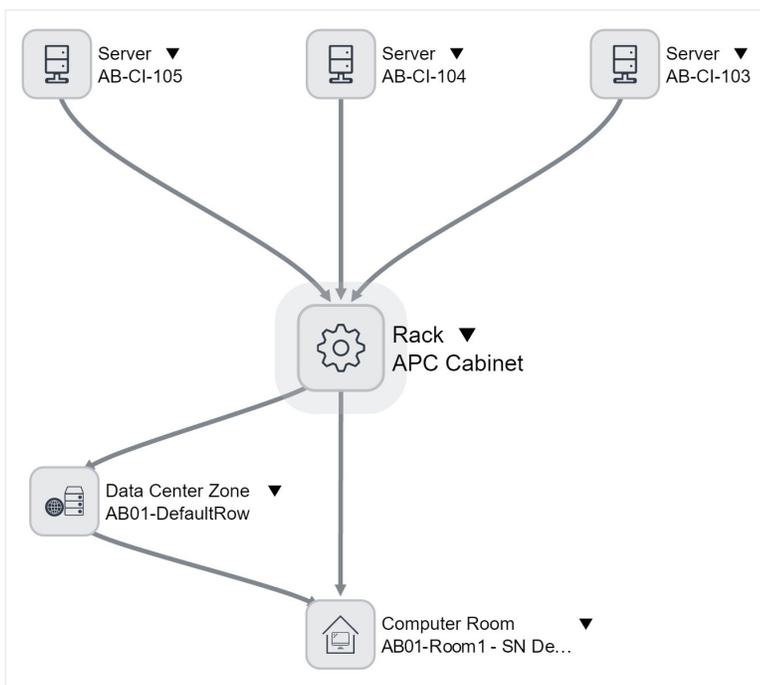


Figure 2: ServiceNow standard relationship example

In the Appendix A: Default ServiceNow CMDB Relationship Types are color coded accordingly. The recommended relationships are green, the ones you might want to swap are yellow, and the red can be eliminated because some other, non-layer-specific relationships can cover them.

# Relationships the Right Way

Before going into the actual setup, let's look at the key tree structures in a data center and their starting points. There are two at the physical level with "location" and "power" meaning the linkages of servers to a rack position and its power source. Each tree starts at the data center level with the physical building and the main power feed.

## The Location Tree

The location tree should represent an easy path from the building to the CMDB-ci element's physical location in the data center. Tier44 EM/8 uses this tree for the visual navigation of a building image, floor plan, and rack elevation view so you can click your way from the building all the way to the server.

ServiceNow has a default relationship type for this called "contains::contained by." It is level independent and can be used for any location type. When used across the tree you end up with something like the picture on the right. The data center at the top, computer room as a child, rack as a child of the computer room and the server and PDUs as children of the rack.

This structure makes it very easy to go up and down the tree as the same ServiceNow relationship type is used and as a result the tree becomes collapsible, meaning the data center also "contains" the servers and PDUs.

With the correct parent/child setup for the relationship, you also can see the arrows pointing in the right direction, something you want to verify with this "Dependency View".

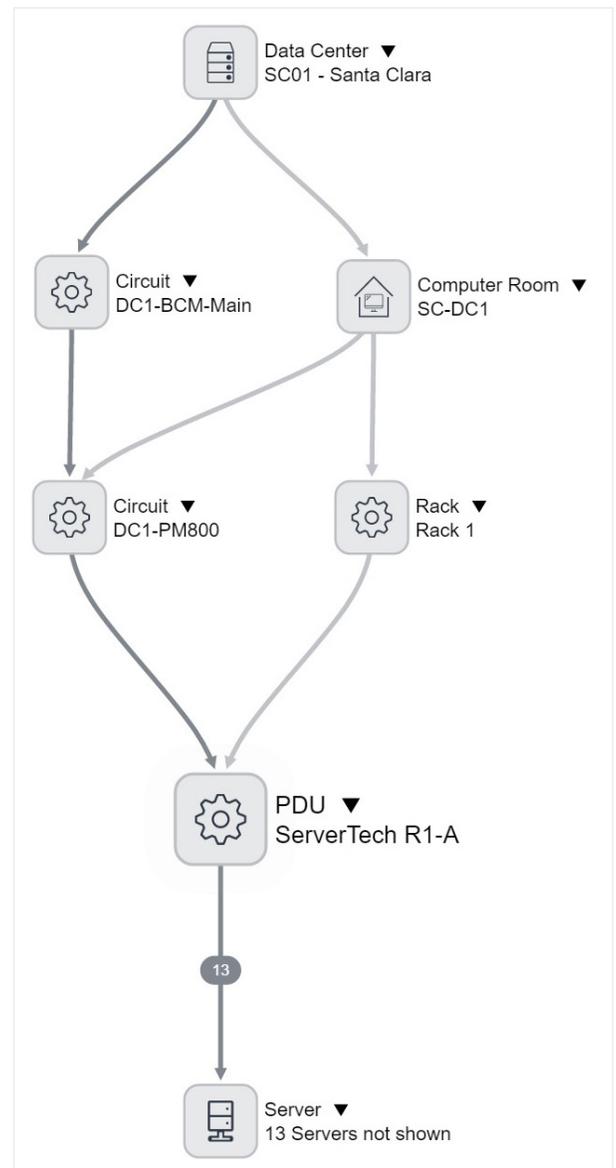


Figure 3: A correctly defined location tree

Looking at the rack in the ServiceNow CMDB, you can also see the defined relationships.

Notice the size of the rack with 40 rack units plus the relationships for the rack. There is a “contained by” Computer Room relationship and multiple “contains” relationships. The dependency view illustrates the relationships.

Turn on EM/8 visualization of the “contains::contained by” relationship and the rack is automatically transformed and can now be recognized by any IT Staff member.

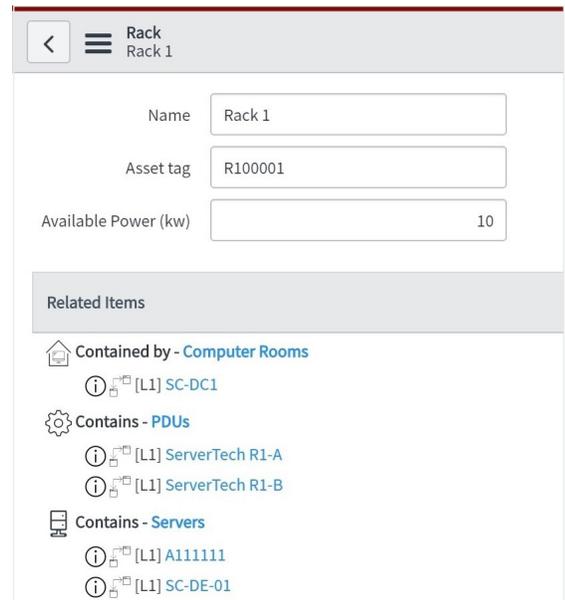


Figure 4: ServiceNow rack configuration details

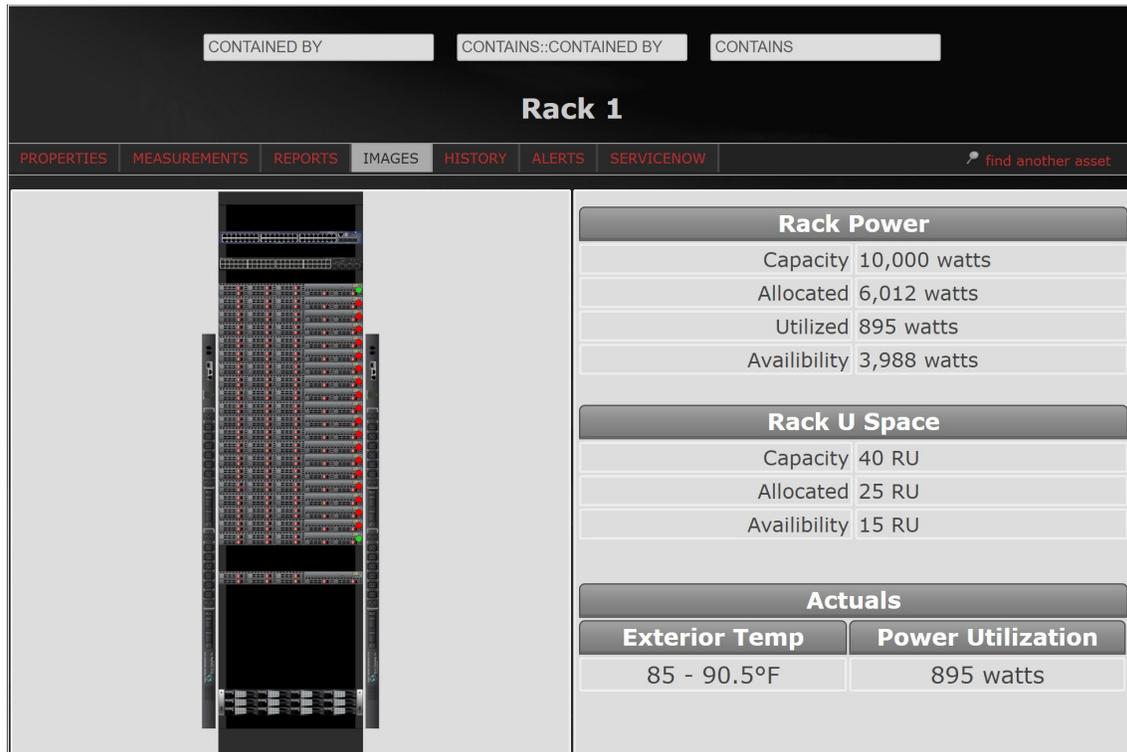


Figure 5: Rack Details

Notice the relationship navigation at the top left towards the parent and right towards the children with the relationship type selector in the middle. You can also see the rack elevation directly from the relationship in ServiceNow. There is other information on the right that is related to contains relationships like available capacity of a rack in terms of space and power plus the demand from the equipment (status and actual power consumption requires monitoring services).

The rack elevation of a specific server is done with the “port” setting of that relationship. Assuming a server to be moved into the 20’s U-position, the port setting would be 20 and for capacity planning the allocation is slot 20 + additional slots based on the u-size of the server or equipment.

Figure 6: Rack Relationship Details

**Note:** Port is defined as an integer, so it does not allow you to specify left/right positions for PDUs or (X, Y) coordinates for markers. We recommend adding a string parameter to the contains relationship to capture this item.

For a server you use the port, in this case U-position 34. For the PDU you would use the t44\_marker\_coordinates that can be “L” or “R” for a PDU image on a rack or (X, Y) pixel coordinates. Here is an example extract from the CMDB\_rel\_ci list:

|                          | Parent                            | Type                                   | Child                           | Port    | t44_marker_coordinates  |
|--------------------------|-----------------------------------|--|---------------------------------|---------|-------------------------|
| <input type="checkbox"/> | <a href="#">Rack 1</a>            | <a href="#">Contains::Contained by</a> | <a href="#">ServerTech R1-B</a> | 0       | R                       |
| <input type="checkbox"/> | <a href="#">Rack 1</a>            | <a href="#">Contains::Contained by</a> | <a href="#">ServerTech R1-A</a> | 0       | L                       |
| <input type="checkbox"/> | <a href="#">RenoDC-F1</a>         | <a href="#">Contains::Contained by</a> | <a href="#">ST-F1-R1-3</a>      | (empty) | 770,330:_small:top-left |
| <input type="checkbox"/> | <a href="#">RenoDC-F1</a>         | <a href="#">Contains::Contained by</a> | <a href="#">ST-F1-R1-2</a>      | (empty) | 770,293:_small:top-left |
| <input type="checkbox"/> | <a href="#">RenoDC-F1</a>         | <a href="#">Contains::Contained by</a> | <a href="#">ST-F1-R1-1</a>      | (empty) | 770,253:_small:top-left |
| <input type="checkbox"/> | <a href="#">RenoDC</a>            | <a href="#">Contains::Contained by</a> | <a href="#">RenoDC-F1</a>       | (empty) | 680,180                 |
| <input type="checkbox"/> | <a href="#">WA01 - MegaCenter</a> | <a href="#">Contains::Contained by</a> | <a href="#">WA01-DH4</a>        | (empty) | 5200,3620:_huge         |
| <input type="checkbox"/> | <a href="#">WA01 - MegaCenter</a> | <a href="#">Contains::Contained by</a> | <a href="#">WA01-DH2</a>        | (empty) | 5200,2040:_huge         |

Figure 7: cmdb\_rel\_ci details

Another example of “contains” relationships that you might want to consider is the relationship of Servers in a cluster. Using the same relationship type extends the tree past the physical level to the application level.

In this case you see a cluster at the top with UNIX servers as children and NAS storage devices as children of the servers.

You could also set up the relationships using CLUSTER members::member of SERVER and SERVER depends on::used by NAS to link the parent to the child. However, doing so breaks the contains tree from the rack to server.

While the parent/child relationship of “member::member of” and “depends on::used by” is correct from the parent/child directional point of view, it does not allow you to collapse the tree, since it’s not a single relationship type for the full tree from CLUSTER to NAS. It would be better to say CLUSTER contains SERVER contains NAS so you can also say CLUSTER contains NAS. We are not suggesting that “contains” is the right term for this particular tree; however, be aware of the possibility of a collapsible structure by using a single relationship type across the full tree.



Figure 8: Server/Cluster Relationship

## The Power Tree

The second main tree in a data center is the “power tree”, typically represented by one or more single line diagrams that show the power distribution from the main infeed/generator to each individual outlet. The goal is a consistent navigation top to bottom with potential measurements and aggregations from measurements.

Similar to the location tree we also look for a consistent relationship top to bottom and ServiceNow made this one easy by defining a “power::powered by” relationship. In this tree the port within the relationship represents an outlet or breaker position by integer.

Here is a sample view into a branch circuit meter on an RPP panel:

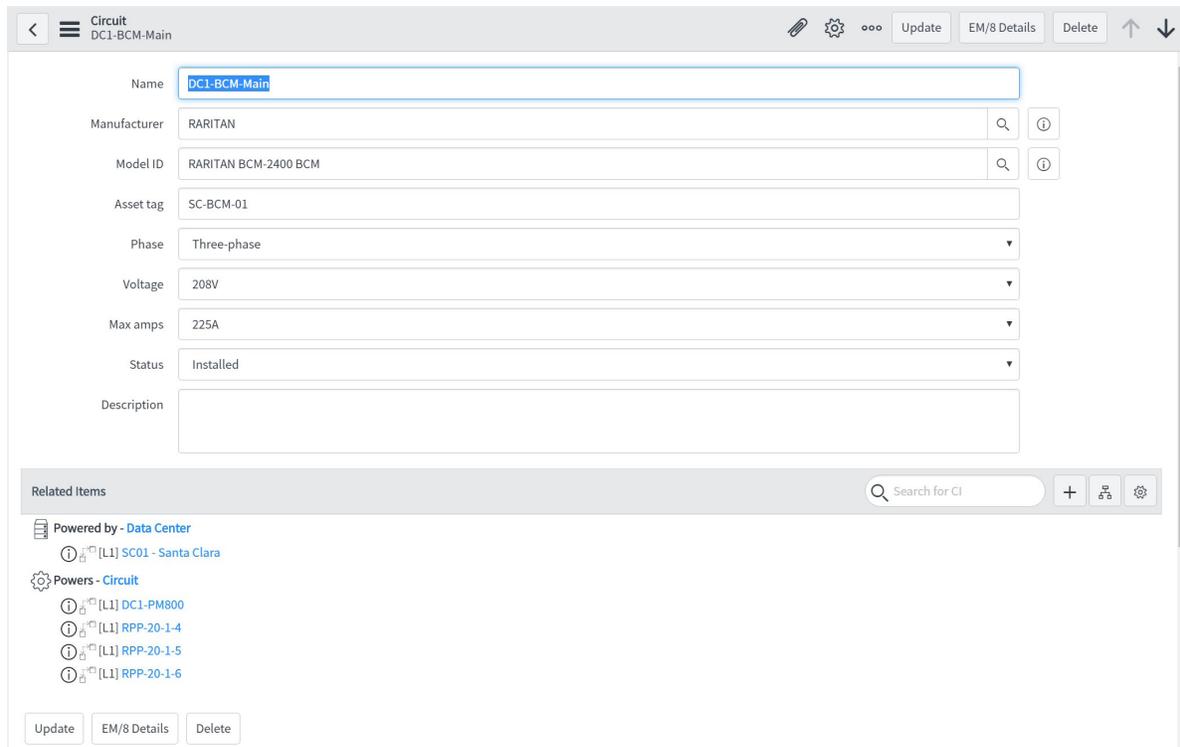


Figure 9: Circuit Details

This particular branch circuit meter has only a couple of relationships defined. There is “powered by”, upstream towards the data center and there are “powers” relationships, meaning circuits going out from the RPP panel.

In a typical data center setup, “powered by” would most likely go to a main power distribution panel, UPS, or other aggregation device. Similarly, there should be more circuits coming out from the branch circuit panel using the “powers” relationship to establish the remaining connector.

When setup properly, this allows you to navigate up and down the power tree from the main feed to the consumer. Here is a dependency matrix of a PDU connected to the branch circuit meter.

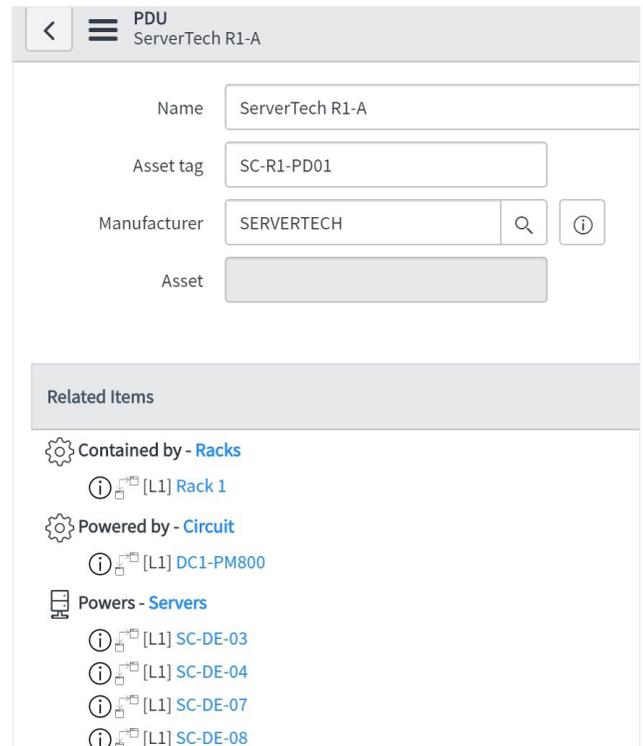


Figure 10: PDU Details

This particular power strip has the “contained by” relationship set to the physical rack and the “powered by” relationship to the branch meter, both serve as child to parent relationships. It also uses “powers” being a parent or provider of power for the servers. The dependency matrix visualizes this relationship tree.

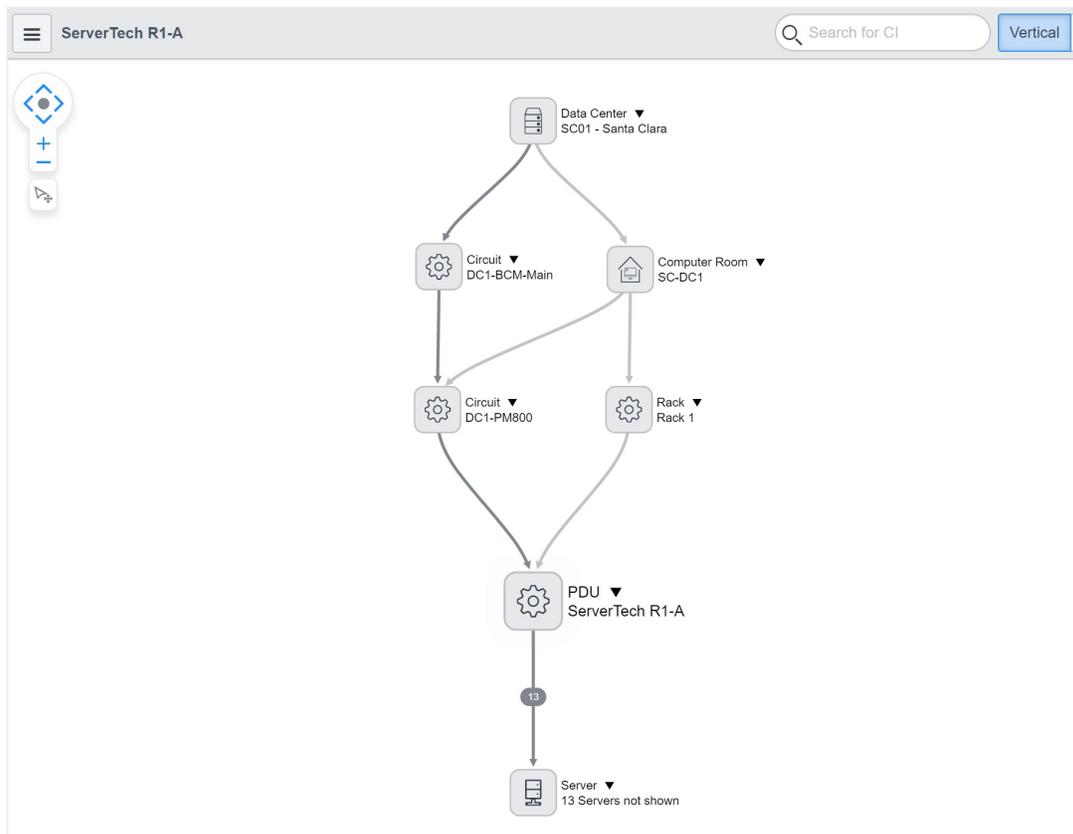


Figure 11: Dependencies and Filters

Don't forget the filters! Since filters are also important. First, we changed the levels to 2 so we go up to the circuit on the branch circuit meter (dark gray line). Second, we disabled the “contains::contained by” relationships so they are only shown in light gray. The result is the power tree on the left and the location tree on the right all coming together at the PDU level.

## The Power and Location Tree Starting Points

For data center owners and operators looking at the overall data center starting points need to be defined. As outlined earlier, the start of the power tree is the single line diagram. For locations it's the data center building and location.

After installing the Tier44 EM/8 application you will see two image fields at the data center detail level. These two fields allow you to upload an image for the building and a single line diagram as the main starting point. Here is the example after EM/8 has been installed and images have been uploaded.

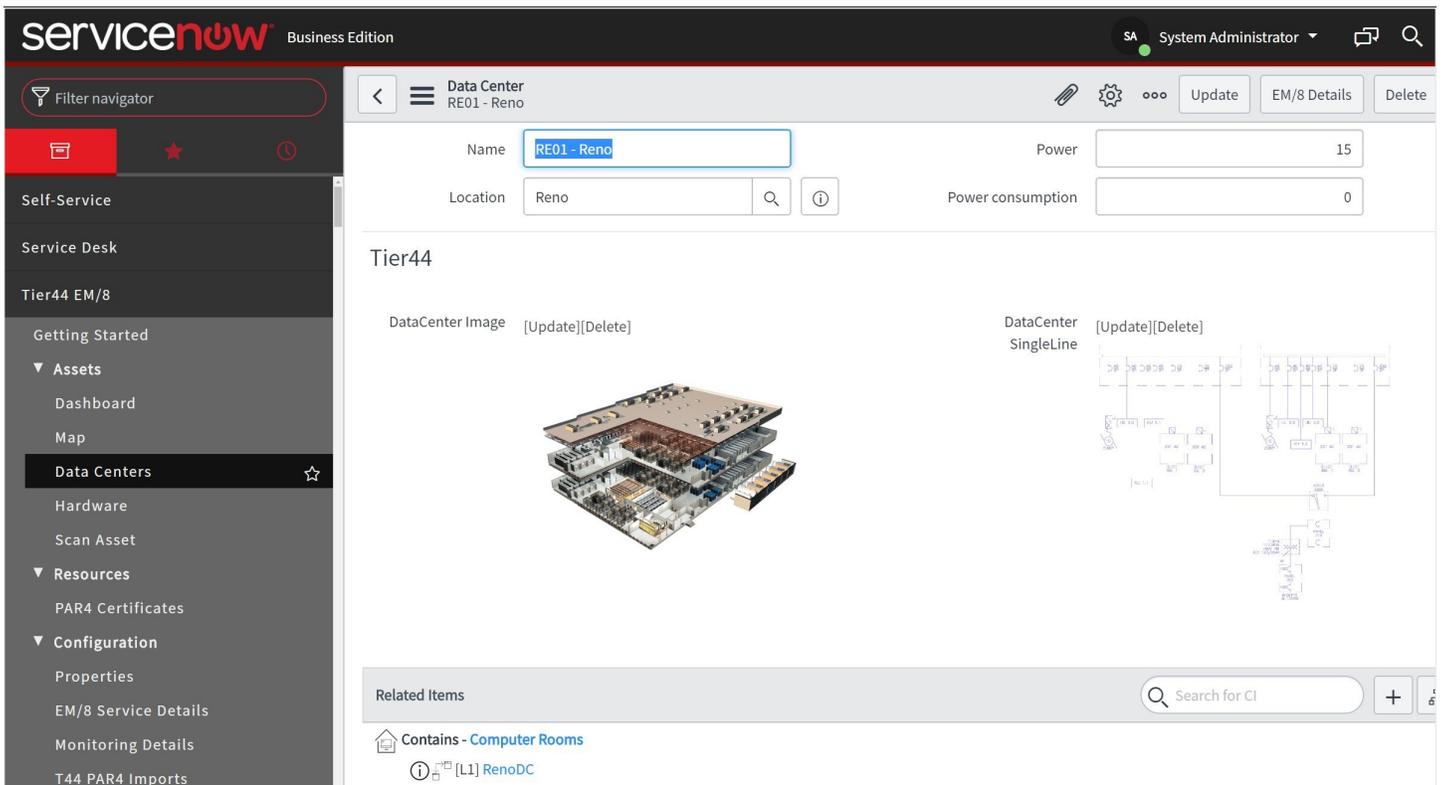
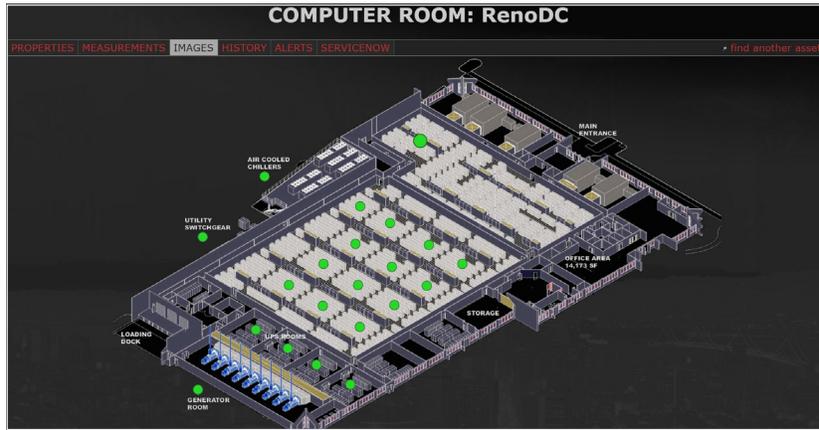


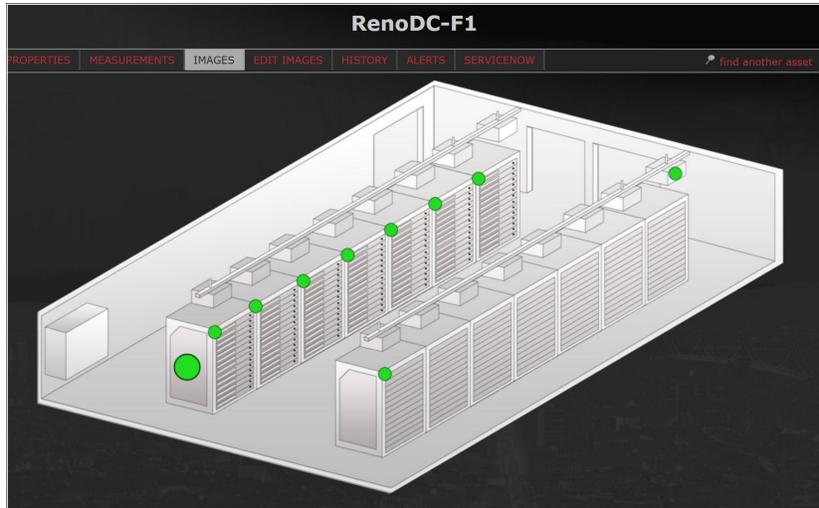
Figure 12: Data Center Images

If you did not install the Tier44 EM/8 application, you can certainly create the image fields on your own, but you won't be able to visually navigate the relationship tree. With EM/8 you can click from one level to the other as long as you stay within the same relationship type, otherwise you have to switch the type at the top of the visualization screens before going further.

Next are some screen sequences showing an end-to-end tree navigation. In this tree, the relationships are indicated by grey arrows. It is possible to traverse from a data center to a room and from the room to a zone (row of racks) or an individual rack. All navigation points are represented by colored dots. Colors can be set based on status or UI design requirements.

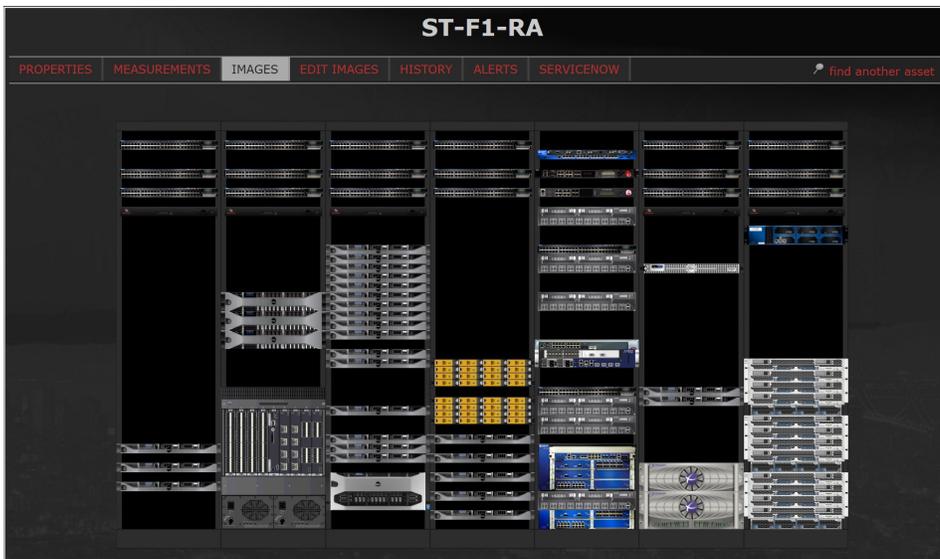


Navigate from Building Floor to Room



Then from Room to Row

Navigate from Room to Individual Rack



Navigate from Row to Individual Rack



**Note:** To visualize the relationships this way you need to make sure the following has been done:

1. Tier44 EM/8 visualization application is installed
2. CMDB CI items have the proper relationships
3. Tree structures share a common type

All views are directly integrated into ServiceNow Geneva and Helsinki releases. Listed is a fully integrated view for illustration purposes with a single line diagram including some navigational relationship markers.

Presented is the familiar ServiceNow navigation on the left and top of the screen. All Tier44 EM/8 screens are embedded and you can switch back and forth from between the visual view and the ServiceNow form at any level.

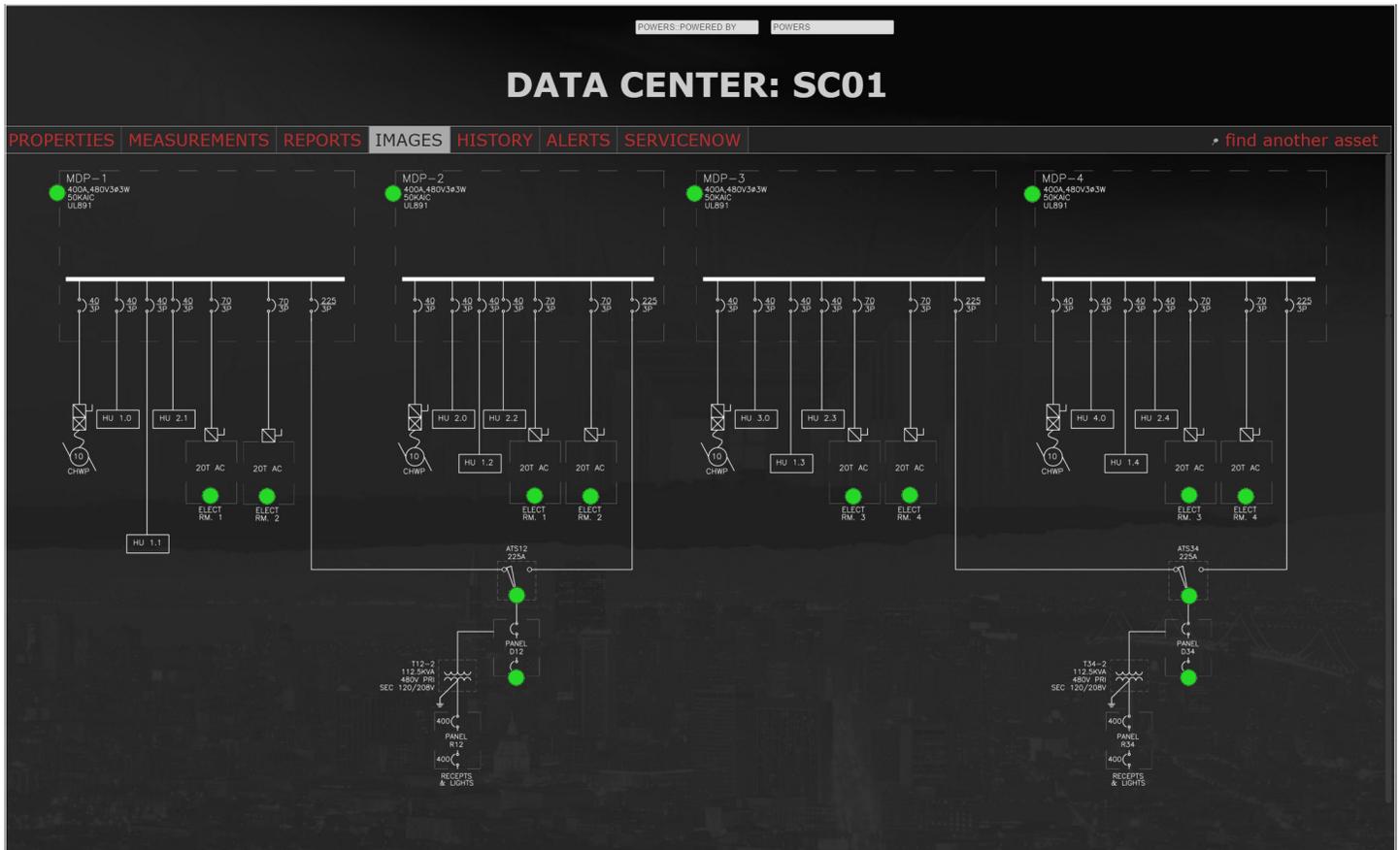


Figure 13: Power Single Line Diagram

You do not have to use Tier44 EM/8; however, we do recommend you setup the relationships in a manner that presents the CMDB in an industry standard meaningful way to reduce confusion for your data center operations staff. That way, when configuring the CMDB, it leads to less confusing relationship structures, minimizes errors on CMDB interconnections and eliminates potential compliance issues by using a clearly defined and updated set of relationships.

# Additional Tree Structures of Interest

For data center owners and operators there are a couple of additional trees of interest when it comes to managing the infrastructure and grouping CMDB\_ci items by business unit, customer or owner. Additional relationships can be defined as needed. We recommend that you limit the number of relationship types and generalize them to cover full tree structures like “contains::contained by” and “powers::powered by.” See Appendix A: Default ServiceNow CMDB Relationship Types for examples of recommended relationship types and some you might want to ignore or even delete to avoid confusion.

## Management Relationships

ServiceNow has a default relationship for “managed by::manages”, but it is backwards. A server or application is managed by something, meaning the server or application is the child while the default relationship requires it to be the parent.

The easiest way to make this relationship type useful is to visit the CMDB\_rel\_type table and simply swap the parent and child. We recommend doing this to establish “manages::managed by” as a relationship type.

When the relationship reversal is complete the ServiceNow dependency matrix is correctly displayed.

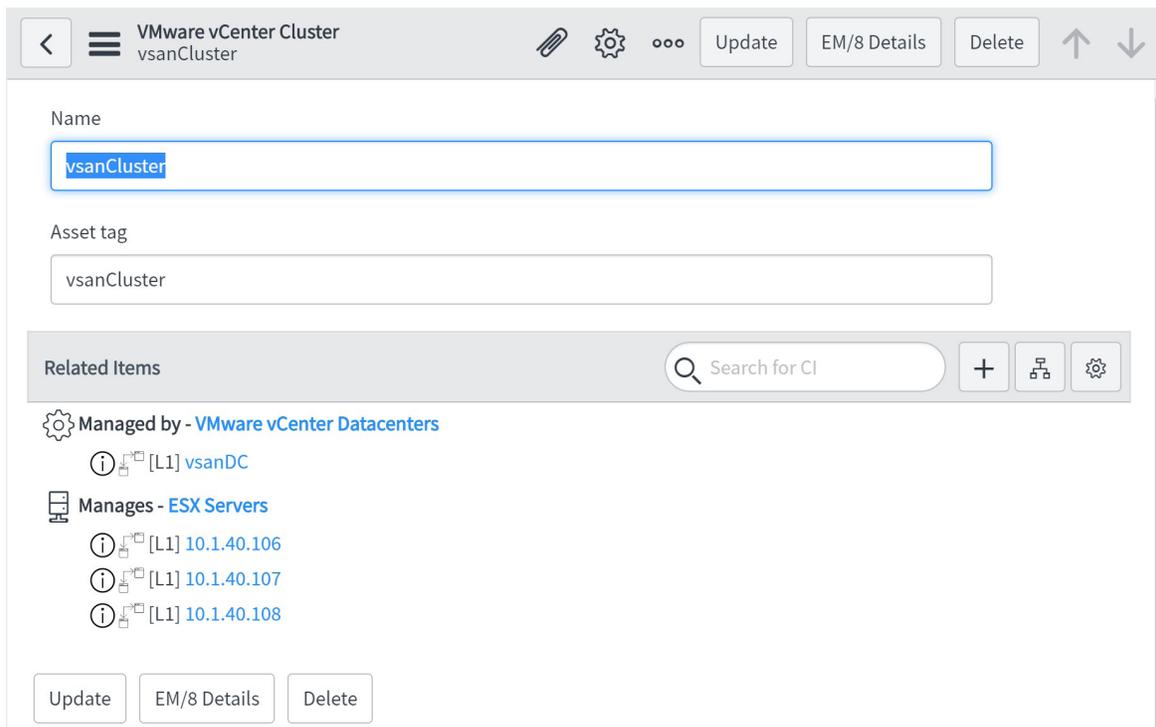


Figure 14: VMware Cluster Configuration

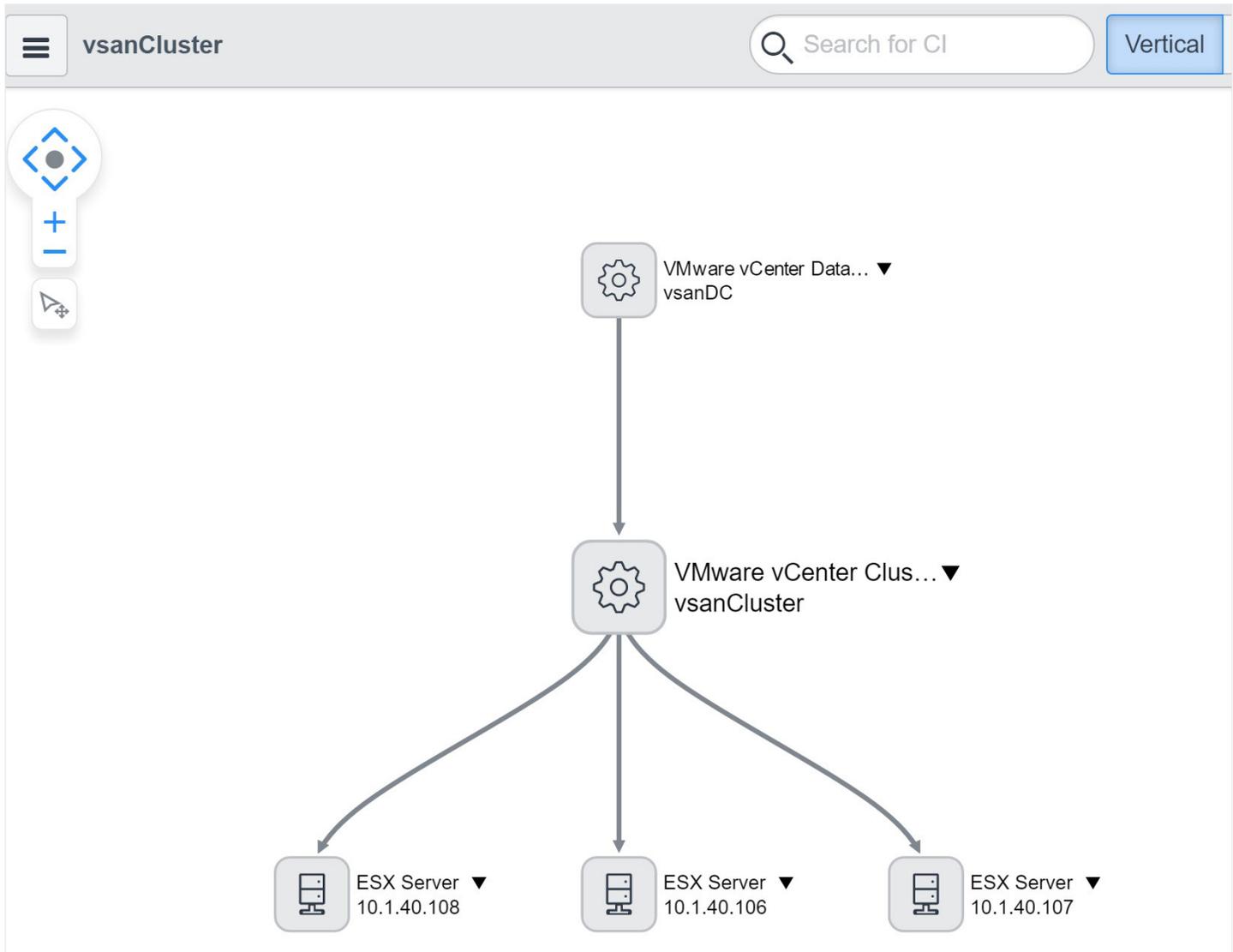


Figure 15: Cluster Dependencies

## Owner Relationships

Another important relationship is around ownership of equipment, circuits, power, applications, etc. for the reasons of building logical groups for chargeback or operating purposes.

ServiceNow has a pre-defined relationship type “uses::used by.” For example, you can define a business unit or customer, abc, who “uses” multiple racks and circuits.

However, be careful as there is also a “used by::uses” relationship type defined where parent items are used by child items. This can be avoided by deleting this relationship type in ServiceNow. Otherwise, you could end up with a customer, abc, being “used by” a rack, circuit, or server.

This particular relationship types works well with “Groups”. Groups can be defined in the CMDB and can use whatever items you want to allocate to them. With EM/8 installed, we can use the “uses” relationship for chargeback and aggregated views.

Here is an example of the group definition with its relationships. In this case the group is a “customer” leasing a rack in a data center with three circuits.

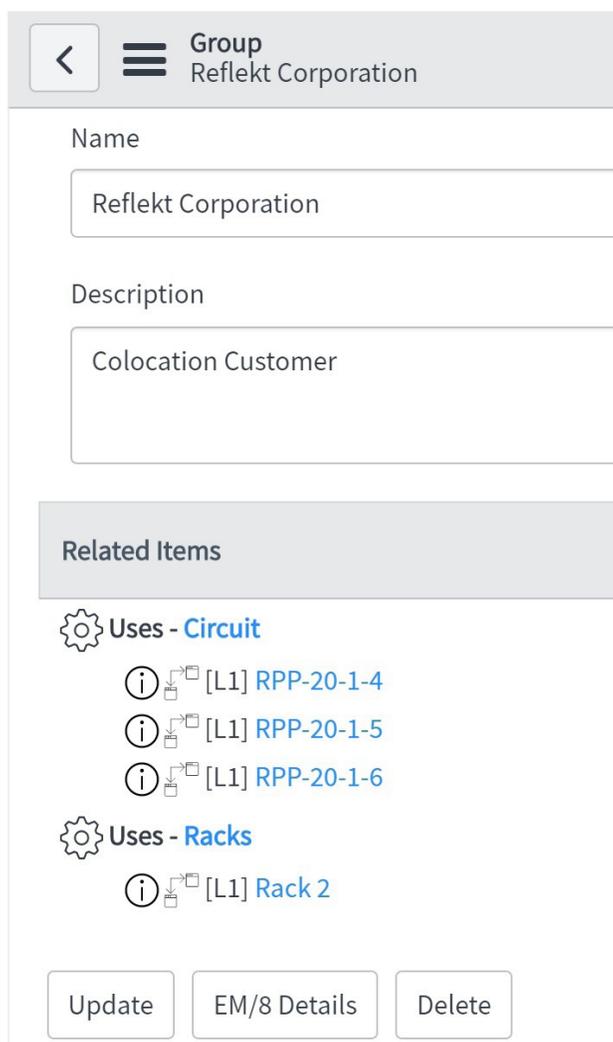


Figure 16: Group Details

The visual representation of these relationships can be aggregated views containing circuit and rack details based on the above example.

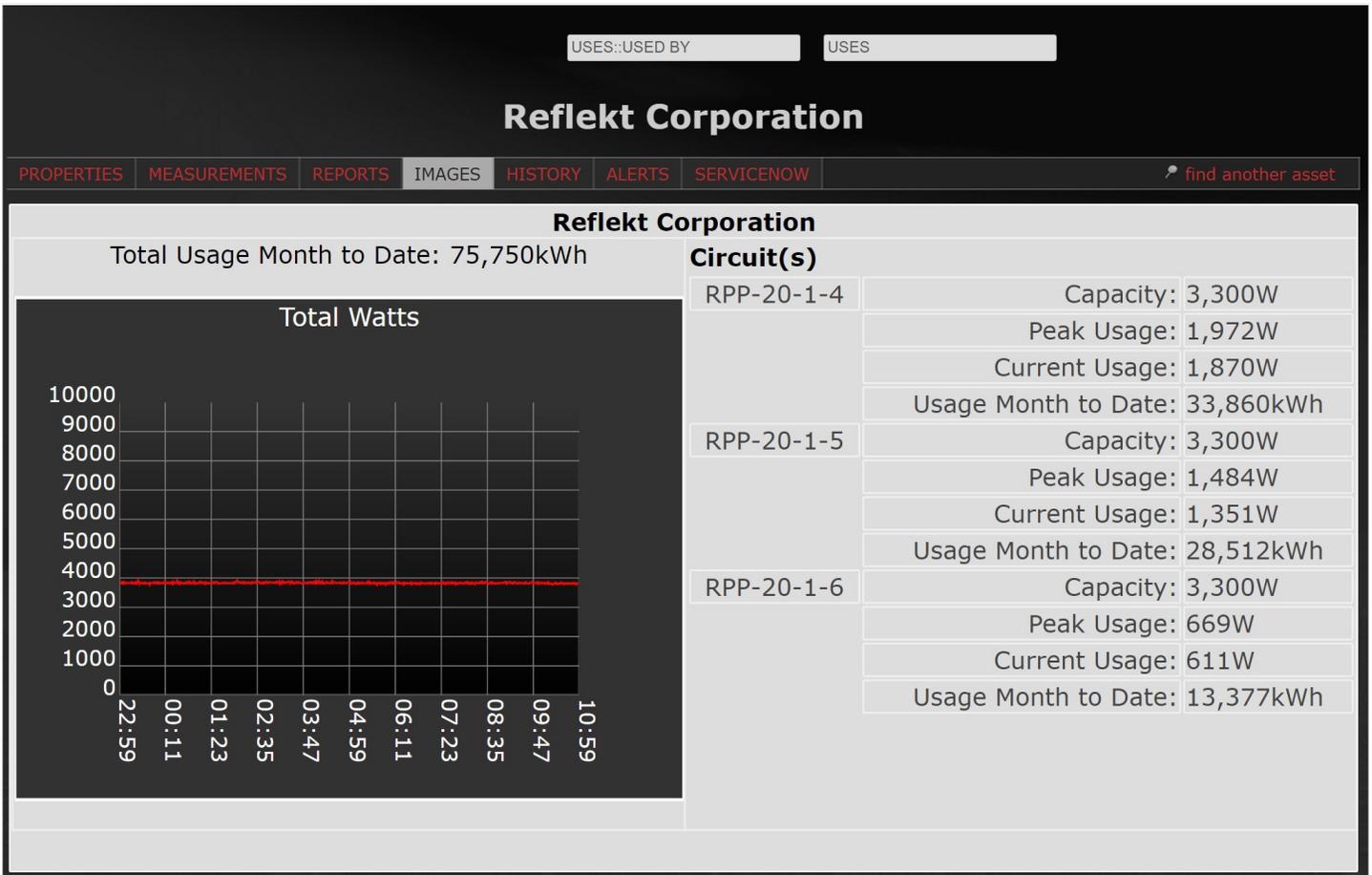


Figure 17: Group Visualization

This particular view shows total power consumption, peak power consumption per circuit, and over-utilization detection.

## Other Relationships

Appendix A: Default ServiceNow CMDB Relationship Types lists the other default relationships as defined by ServiceNow. Some of these relationships are for other, non-data center specific purposes. You can decide which filters you want to use. We recommend verifying them based on their parent/child use and also the uniqueness relative to a complete tree. The relationships marked in green are acceptable in their current context. We recommend that you reverse the relationships marked in yellow. Relationships in red are duplicates and should be considered for removal or deletion.

## About CDI LLC

CDI LLC was founded in 1995, with corporate headquarters in Teterboro, NJ, as well as office locations in New York City, Boston, Philadelphia, Charlotte and Atlanta. As one of the nation's top 500 business IT solution providers, the firm architects, deploys and manages multiplatform hybrid IT solutions, including traditional IT, public, private and hybrid clouds, to a wide variety of industries. CDI LLC continually focuses on achieving client satisfaction by developing and implementing comprehensive and innovative technology solutions that enhance day-to-day business automation and workflow processes.

For more information, visit [cdillc.com](http://cdillc.com) or call 1-800-234-5531. Follow us for continual coverage on Twitter [@cdillc](https://twitter.com/cdillc) or YouTube.

## About Tier44 Technologies, Inc.

Tier44 Technologies, a ServiceNow and VMware Technology Alliance partner, provides data center management solutions for visualization, monitoring and managing IT and facility components inside and across data centers. Tier44's patented and award winning solutions will help data centers to visualize floors and racks, monitor physical and virtual applications, power, temperature and IT utilization, optimize IT capacity and ultimately cut power consumption in half while increasing application reliability by enabling applications to be shifted between data centers dynamically.

Tier44's solutions are based on and certified by ServiceNow and are available in the ServiceNow Store. They extend ServiceNow, use the ServiceNow platform and provide data center specific extensions to increase the value of ServiceNow for data center owners and operators. The solutions are also certified by VMware and are available in the VMware solution exchange.

## Default ServiceNow CMDB Relationship Types

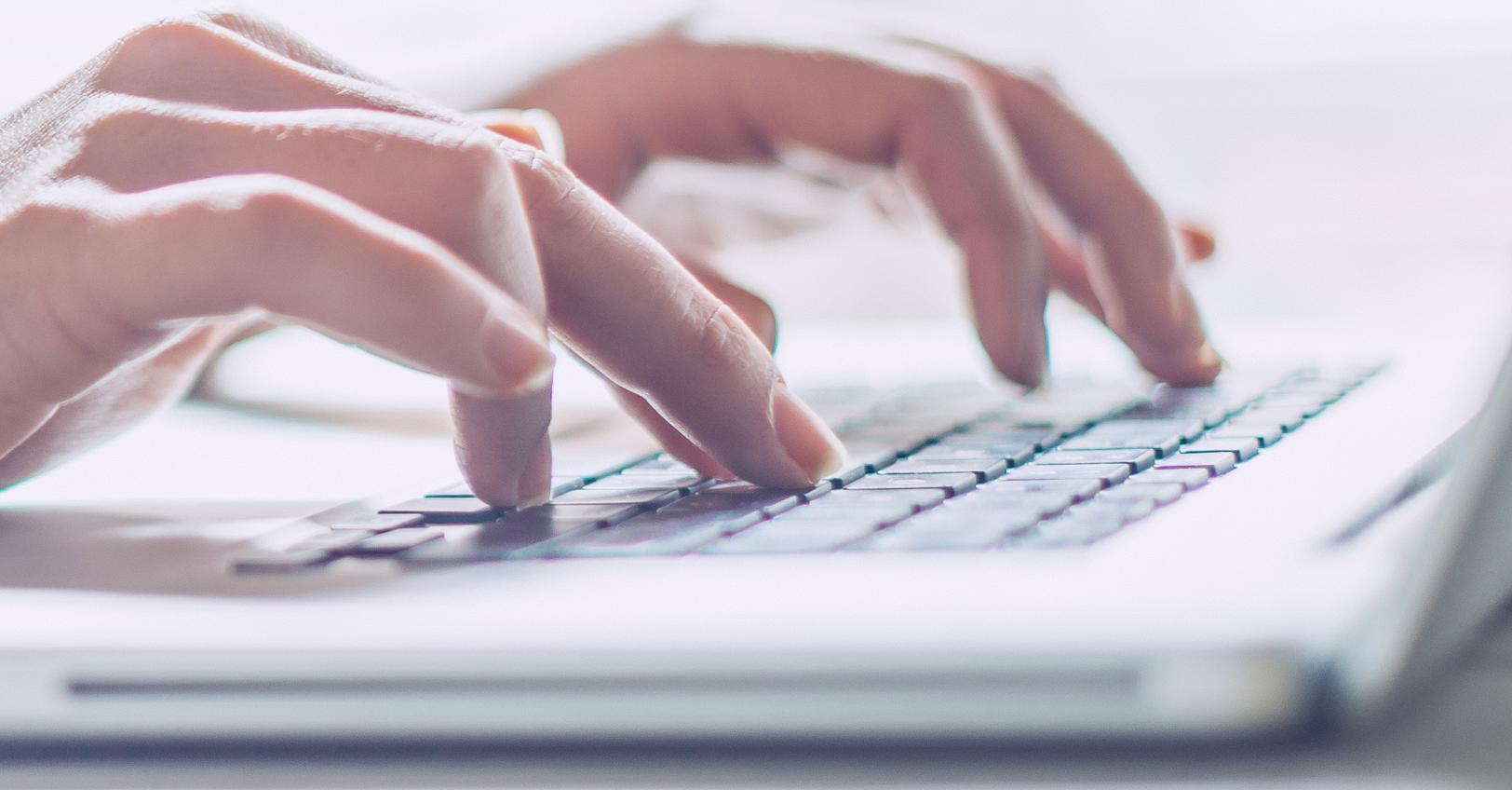
Green relationships are ready to use, you might want to flip yellow relationships and ignore red ones.

| PARENT DESCRIPTOR      | CHILD DESCRIPTOR         |
|------------------------|--------------------------|
| Applicative Flow To    | Applicative Flow From    |
| Cluster of             | Cluster                  |
| Connected by           | Connects                 |
| Connects to            | Connected by             |
| Consumed by            | Consumes                 |
| Consumes               | Consumed by              |
| Contains               | Contained by             |
| Cools                  | Cooled by                |
| Defines resources for  | Gets resources from      |
| Depends on             | Used by                  |
| Distributed by         | Distributes              |
| DR provided by         | Provides DR for          |
| Exchanges data with    | Exchanges data with      |
| Feeds                  | Fed by                   |
| From template          | Template for             |
| Hosted on              | Hosts                    |
| Implement End Point To | Implement End Point From |
| In Rack                | Rack contains            |
| Instantiates           | Instance of              |
| Instantiates           | Instantiated by          |
| IP Connection          | IP Connection            |
| Located in Zone        | Zone contains            |
| Located in             | Contains Room            |
| Located in             | Houses                   |
| Managed by             | Manages                  |
| Members                | Member of                |
| Powers                 | Powered by               |
| Provided By            | Provides                 |
| Provides storage for   | Stored on                |
| Provides               | Provided by              |
| Receives data from     | Sends data to            |
| Registered on          | Has registered           |
| Runs on                | Runs                     |
| Use End Point To       | Use End Point From       |
| Used by                | Uses                     |
| Uses                   | Used by                  |
| Virtualized by         | Virtualizes              |

Note: From CMDB\_rel\_type\_list

## Contact CDI LLC or Tier44 Technologies, Inc. to get started

To learn more about how your organization can solve these structural team relationship challenges in ServiceNow CMDB please visit either the CDI website at [www.cdillc.com](http://www.cdillc.com) or call 1-800-234-5531 or Tier44 Technologies at [www.tier44.com](http://www.tier44.com) or call +1.408.980.9500.



© 2014-2016 Tier44 Technologies, Inc.. All rights reserved.

Tier44, EM/8, PAR4, Demand Balancer, Supply Balancer, Holistic Power Management are registered trademarks of Tier44 Technologies, Inc. All other product names, company names, logos, and trademarks are used herein for identification purposes only and are the property of their respective companies.

[www.tier44.com](http://www.tier44.com)