

Was verrät Ad-Hoc-Reuse in unserer Anwendungslandschaft über unsere Organisation?

Und was nützt uns das?

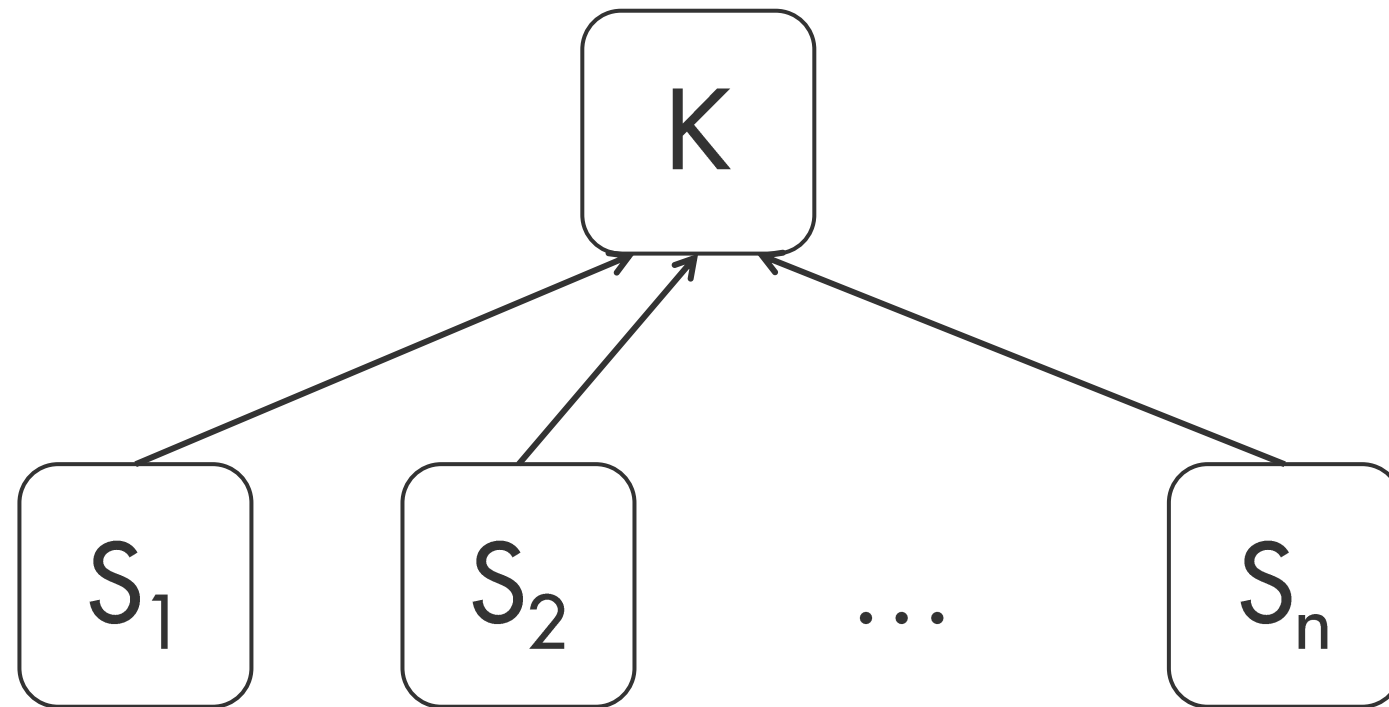
TUM



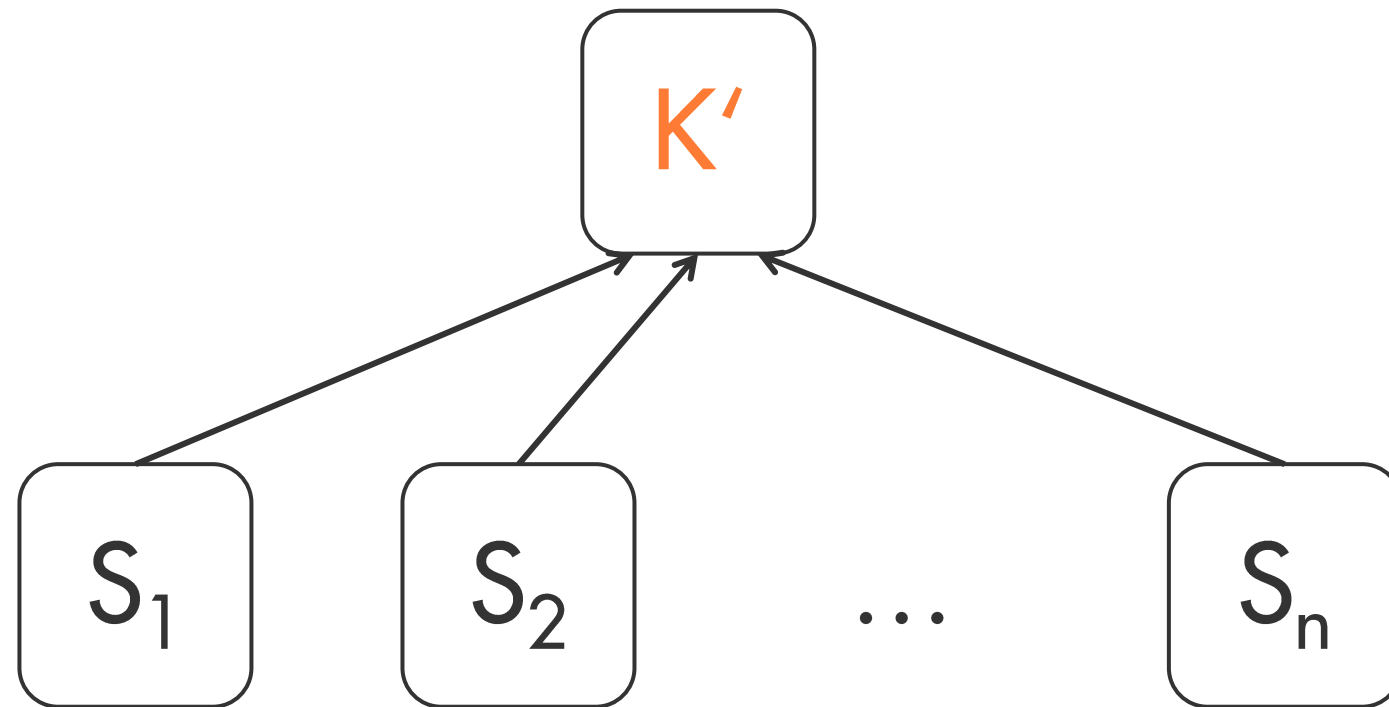
CQSE



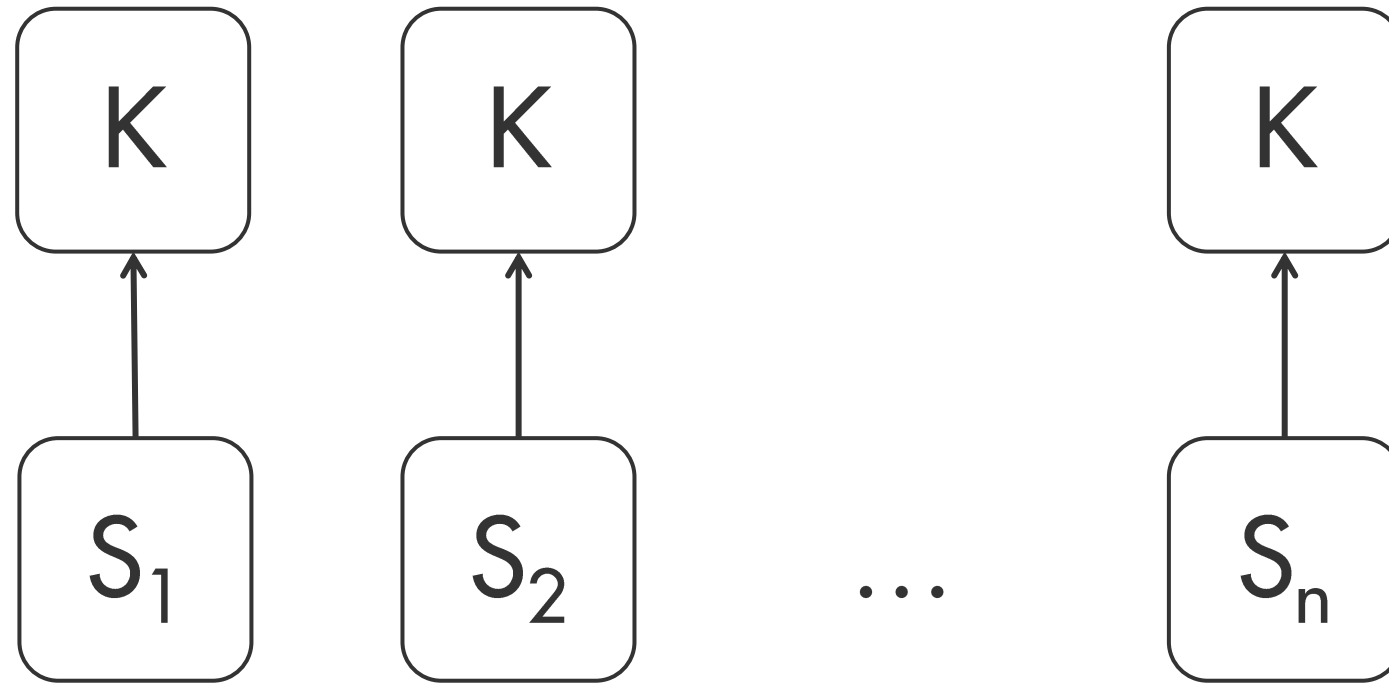
Systematischer Reuse



Systematischer Reuse



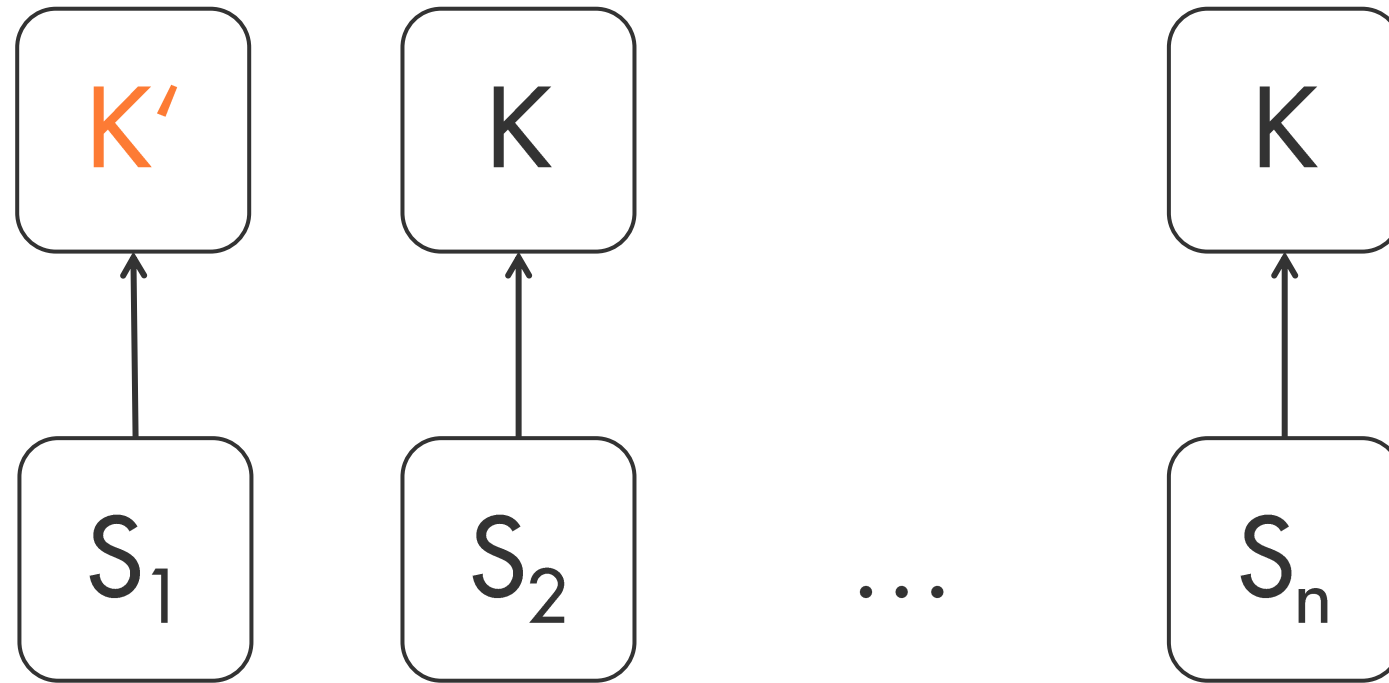
Ad-Hoc Reuse (Copy & Paste)



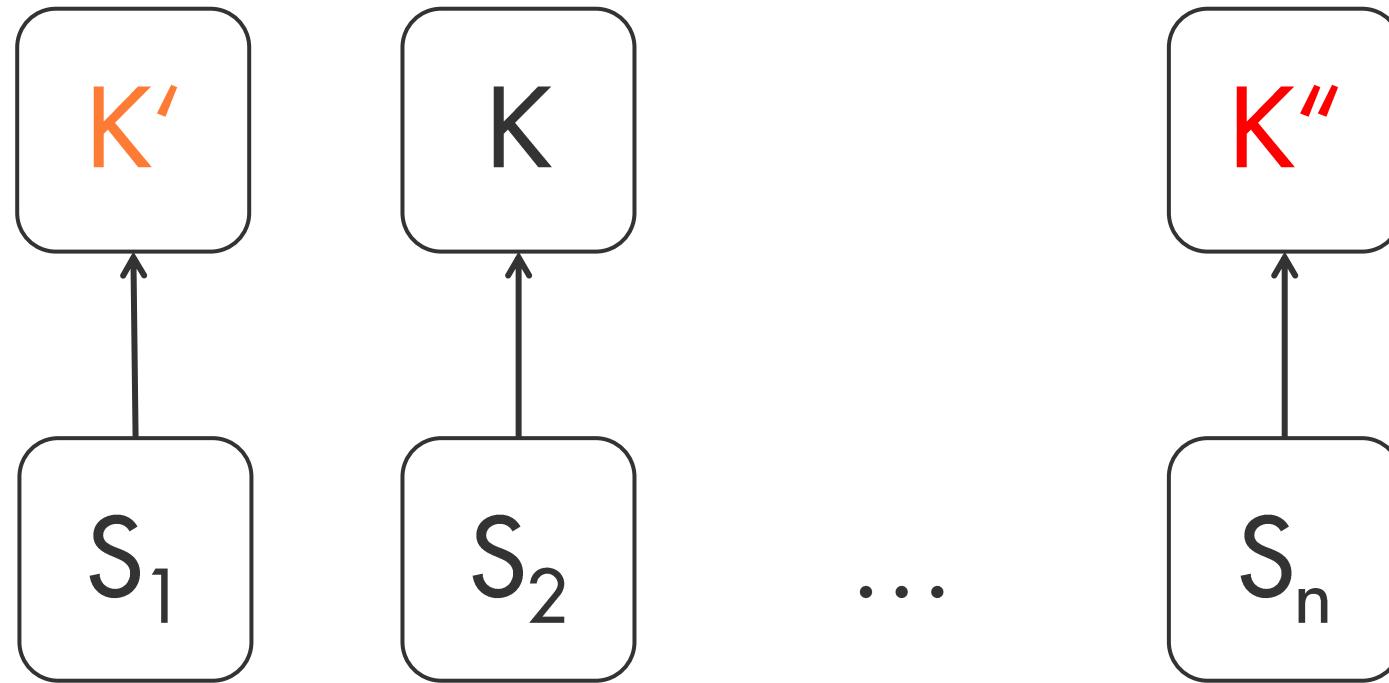
```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

Ad-Hoc Reuse (Copy & Paste)



Ad-Hoc Reuse (Copy & Paste)




```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

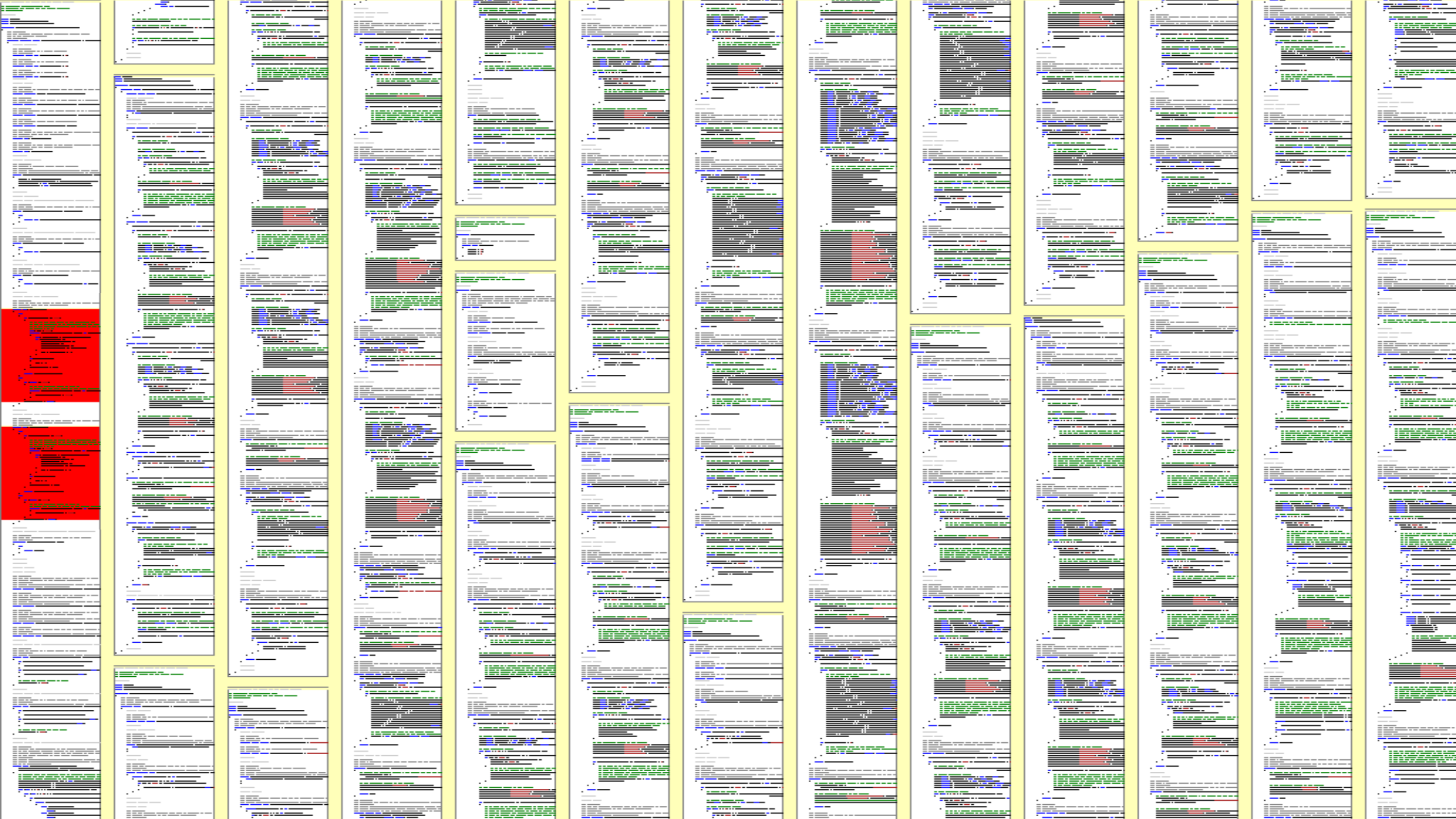
**Wieviel Ad-Hoc Reuse
haben wir bei uns?**

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

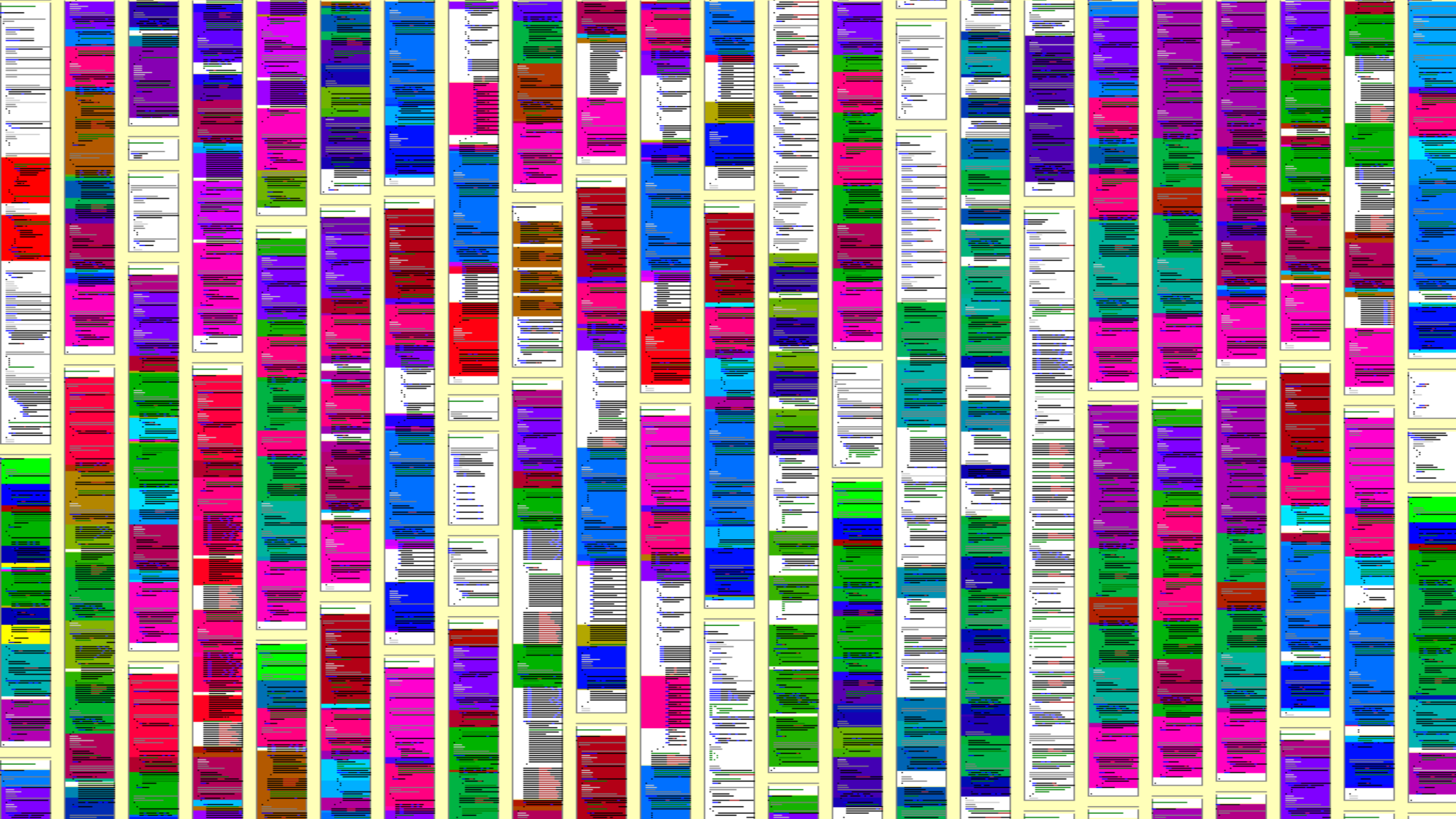
```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

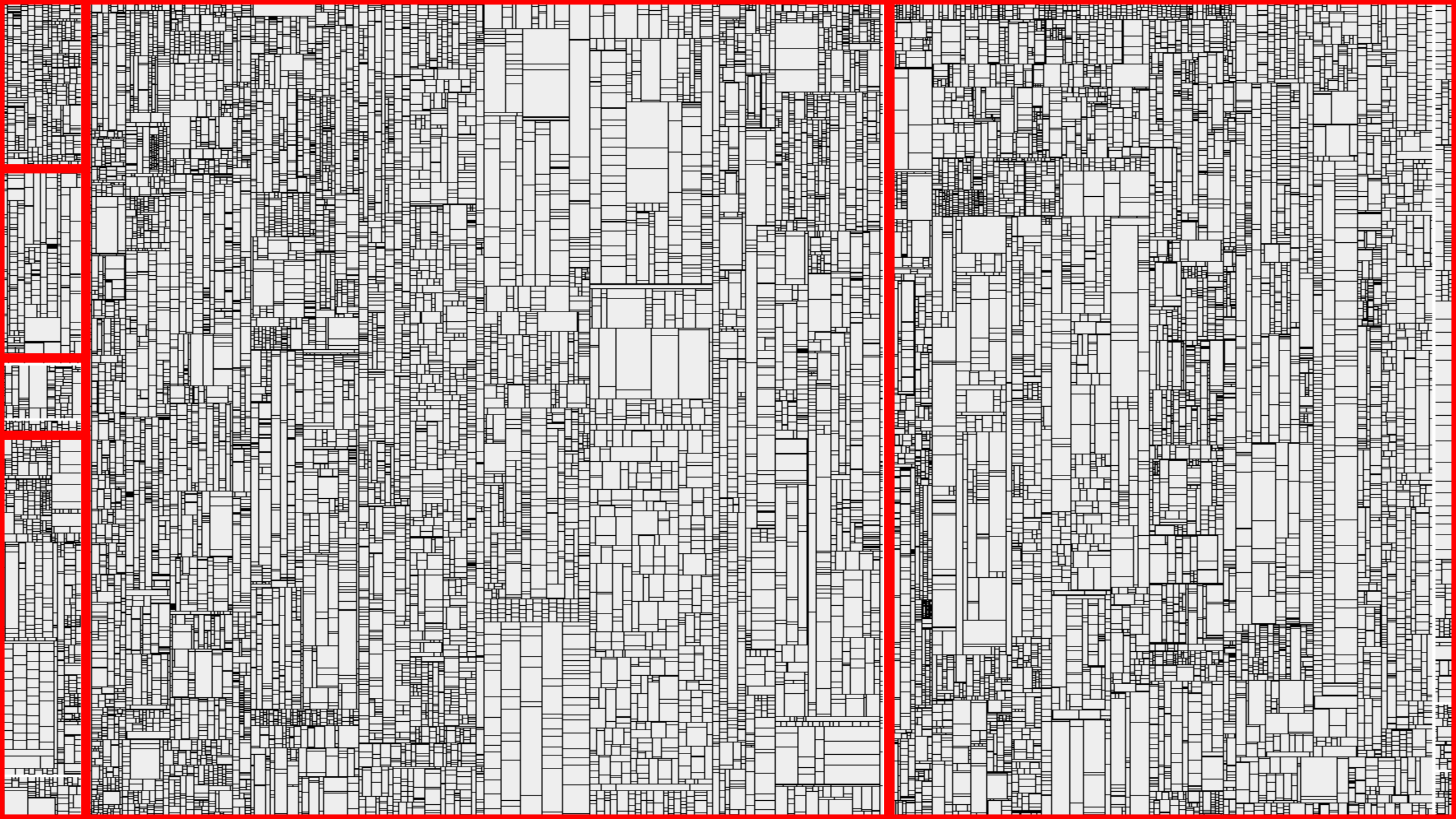
```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```



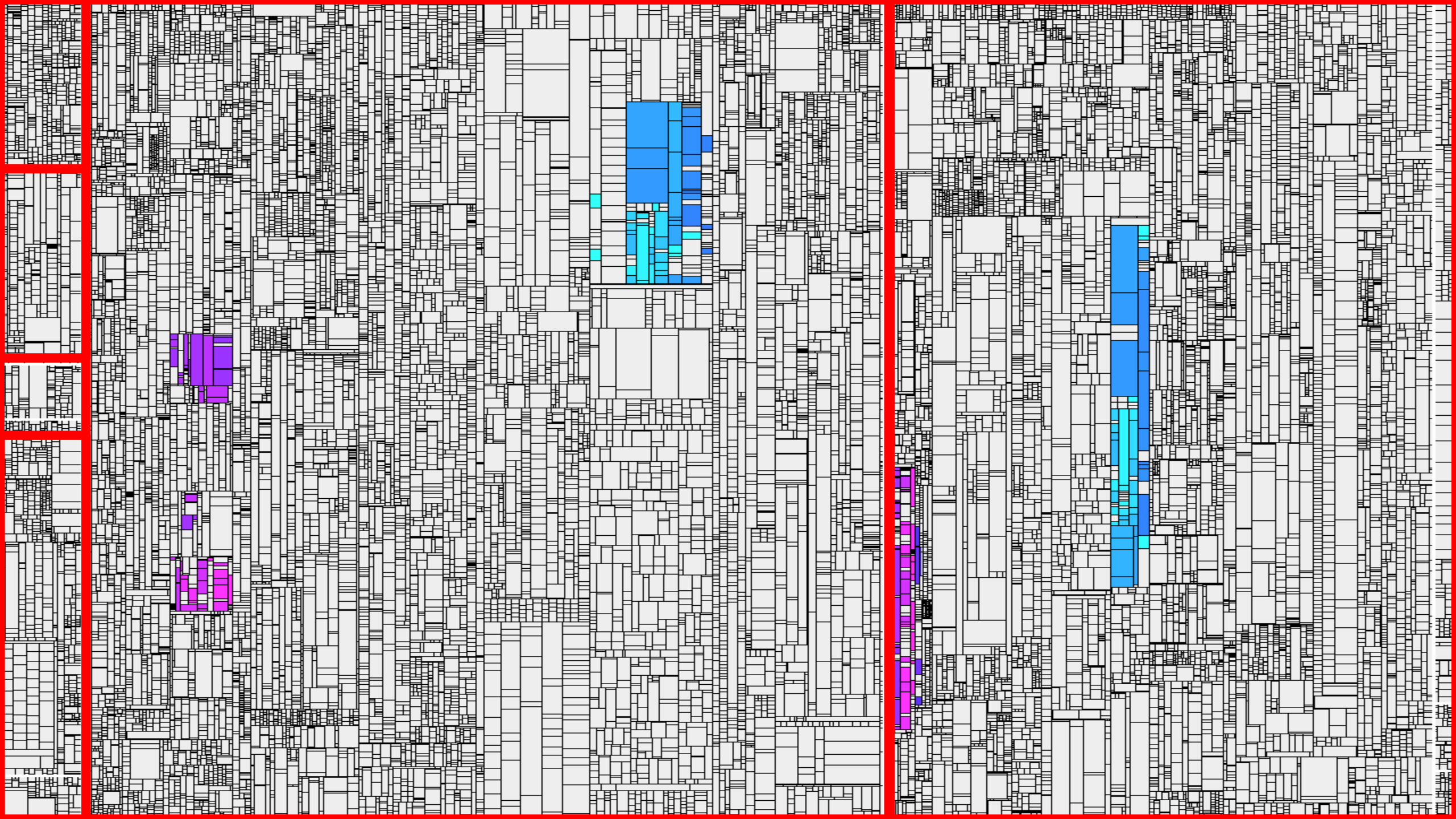


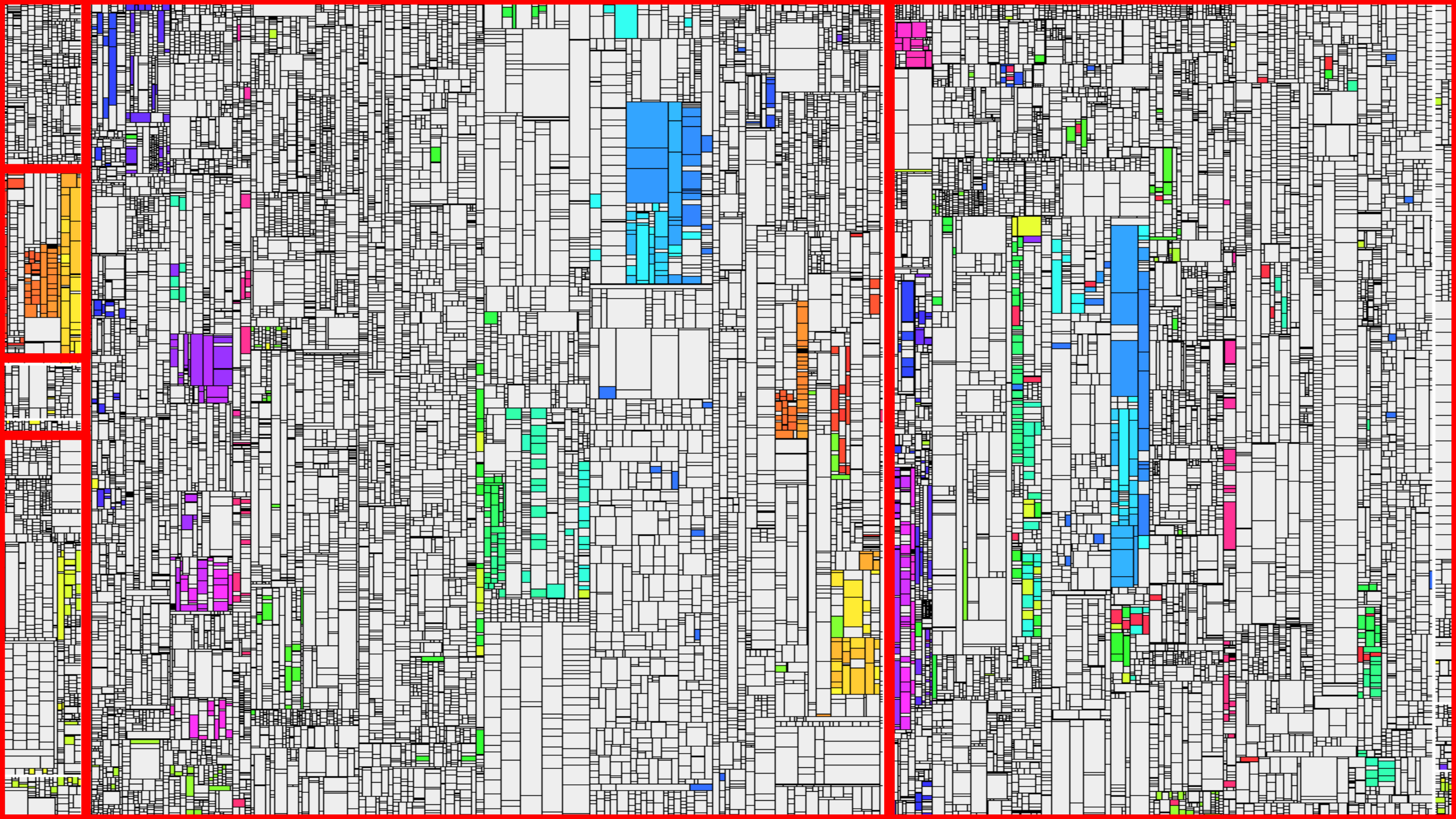












Schafft das Probleme?

```

/// <param name="authority">Die Zuweisung die deaktiviert werden soll</param>
void IAuthorityListManager.DeactivateAuthority(DecMemoIdentifier decMemoId,
{
    this.delegateManager.DeactivateAuthority(decMemoId, authorityList as Autho
}

/// <summary>
/// Führt die Laufliste fort (geht zur nächsten Zuweisung über wenn die aktuelle Z
/// </summary>
/// <param name="decMemoId">Identifikator der Entscheidungsvorlage</param>
/// <param name="authorityList">Die Laufliste die fortgeführt werden soll</para
/// <returns>Die nach dem Fortführen aktive Zuweisung der Laufliste</returns>
IAuthorityAssignment IAuthorityListManager.Proceed(DecMemoIdentifier decMemoId,
{
    if (!this.CheckCurrentUserMayProceed(authorityList as AuthorityList))
    {
        throw new AuthorityListException(Error_27.CurrentUserMayNotProceedAut
    }
    if (authorityList.State == AuthorityListState.InProgress)
    {
        DTOComplex decMemoData = this.GetDecMemo(decMemoId, Currency.Neu
        ((IDecMemoState)this).SubmitDecMemo(decMemoId, decMemoData);
        IAuthorityAssignment newActiveAssignment = this.delegateManager.Proceed
        return newActiveAssignment;
    }
    else
    {
        return this.delegateManager.Proceed(decMemoId, authorityList as Authority
    }
}
...

```

```

void IAuthorityListManager.DeactivateAuthority(DecMemoIdentifier decMemoId,
{
    this.delegateManager.DeactivateAuthority(decMemoId, authorityList as
}

/// <summary>
/// Führt die Laufliste fort (geht zur nächsten Zuweisung über wenn die ak
/// </summary>
/// <param name="decMemoId">Identifikator der Entscheidungsvorlage<
/// <param name="authorityList">Die Laufliste die fortgeführt werden soll
/// <returns>Die nach dem Fortführen aktive Zuweisung der Laufliste</re
IAuthorityAssignment IAuthorityListManager.Proceed(DecMemoIdentifier decMemoId,
{
    if (!this.CheckCurrentUserMayProceed(authorityList as AuthorityList))
    {
        throw new AuthorityListException(Error_27.CurrentUserMayNotProceedAut
    }
    if (authorityList.State == AuthorityListState.InProgress)
    {
        DTOComplex decMemoData = ((ICedentDecMemoStore)this).GetCed
        ((IDecMemoState)this).SubmitDecMemo(decMemoId, decMemoData);
        IAuthorityAssignment newActiveAssignment = this.delegateManager
        base.CommitTransaction();
        return newActiveAssignment;
    }
    else
    {
        return this.delegateManager.Proceed(decMemoId, authorityList as A
    }
}
...

```

```

if i_kart-life_value < 0.
  write '50'          to it_down-newbs.
  i_kart-life_value = i_kart-life_value * -1.
  write i_kart-life_value to it_down-wrbtr
                        currency i_kost_ges-currency.

endif.
it_down-xblnr      = w_xblnr.
it_down-zuonr     = w_zuonr.
it_down-sgtxt     = w_sgtxt.
it_down-kostl     = i_kost_ges-fi_life_cost_center.
it_down-zzvber    = '10'.
it_down-bukrs     = i_kost_ges-booking.
*
if it_down-bukrs ne '0001'.
  write 'SA'        to it_down-blart.
  clear it_down-zzvber.
endif.
*
replace all occurrences of '.' in it_down-wrbtr with ''.
append it_down.
endloop.
*
describe table it_down lines l_it_down.
it_down-blkz = 'E'.
modify it_down index l_it_down transporting blkz.
endloop.
*
*--- Non-Life Quartal
loop at i_kost_ges where period_type = 'q' and
                        non_life_value ne 0.
                        and fehler ne 'X'.
*
if i_kost_ges-fibu_type = 'Y'.
  d_fibu_type = text-k01.

```

```

if i_kart-non_life_value < 0.
  write '50'          to it_down-newbs.
  i_kart-non_life_value = i_kart-non_life_value * -1.
  write i_kart-non_life_value to it_down-wrbtr
                        currency i_kost_ges-currency.

endif.
*
it_down-xblnr      = w_xblnr.
it_down-zuonr     = w_zuonr.
it_down-sgtxt     = w_sgtxt.
it_down-kostl     = i_kost_ges-fi_non_life_cost_center.
it_down-zzvber    = '10'.
it_down-bukrs     = i_kost_ges-booking.
*
if it_down-bukrs ne '0001'.
  write 'SA'        to it_down-blart.
  clear it_down-zzvber.
endif.
*
append it_down.
*
endloop.
describe table it_down lines l_it_down.
it_down-blkz = 'E'.
modify it_down index l_it_down transporting blkz.
endloop.
*
*--- Monatlich Health
loop at i_kost_ges where period_type = 'm' and
                        health_value ne 0.
                        fehler ne 'X'.
*
if i_kost_ges-fibu_type = 'Y'.
  d_fibu_type = text-k01.

```

- Dashboard
- Activity
- Findings
- Metrics
- Tests
- Issues
- Tasks
- Architecture
- Delta
- Projects
- System
- Admin

jenkins/test/src/main/java/org/jvnet/hudson/test/HudsonTestCase.java

(revision 12d96a56...)

```

if (lhs==null && rhs==null) return;
if (lhs==null) fail("lhs is null while rhs="+rhs);
if (rhs==null) fail("rhs is null while lhs="+lhs);

Constructor<?> lc = findDataBoundConstructor(lhs.getClass());
Constructor<?> rc = findDataBoundConstructor(rhs.getClass());
assertEquals("Data bound constructor mismatch. Different type?",lc,rc);

List<String> primitiveProperties = new ArrayList<String>();

String[] names = ClassDescriptor.loadParameterNames(lc);
Class<?>[] types = lc.getParameterTypes();
assertEquals(names.length,types.length);
for (int i=0; i<types.length; i++) {
    Object lv = ReflectionUtils.getPublicProperty(lhs, names[i]);
    Object rv = ReflectionUtils.getPublicProperty(rhs, names[i]);

    if (Iterable.class.isAssignableFrom(types[i])) {
        Iterable lcol = (Iterable) lv;
        Iterable rcol = (Iterable) rv;
        Iterator ltr,rtr;
        for (ltr=lcol.iterator(), rtr=rcol.iterator(); ltr.hasNext() && rtr.hasNext();){
            Object litem = ltr.next();
            Object ritem = rtr.next();

            if (findDataBoundConstructor(litem.getClass())!=null) {
                assertEqualsDataBoundBeans(litem,ritem);
            } else {
                assertEquals(litem,ritem);
            }
        }
        assertFalse("collection size mismatch between "+lhs+" and "+rhs, ltr.hasNext() ^
    } else
    if (findDataBoundConstructor(types[i])!=null || (lv!=null && findDataBoundConstructo
        // recurse into nested databound objects
        assertEqualsDataBoundBeans(lv,rv);
    } else {
        primitiveProperties.add(names[i]);
    }
}

// compare shallow primitive properties
if (!primitiveProperties.isEmpty())
    assertEqualsBeans(lhs,rhs,Util.join(primitiveProperties,""));

*
Makes sure that two collections are identical via {@link #assertEqualDataBoundBeans(Object
/
public void assertEqualsDataBoundBeans(List<?> lhs, List<?> rhs) throws Exception {
    assertEquals(lhs.size(), rhs.size());
}
    
```

jenkins/test/src/main/java/org/jvnet/hudson/test/JenkinsRule.java

(revision 3909f5ac...)

```

if (lhs==null && rhs==null) return;
if (lhs==null) fail("lhs is null while rhs="+rhs);
if (rhs==null) fail("rhs is null while lhs="+lhs);

Constructor<?> lc = findDataBoundConstructor(lhs.getClass());
Constructor<?> rc = findDataBoundConstructor(rhs.getClass());
assertThat("Data bound constructor mismatch. Different type?", (Constructor)rc, is((Cons

List<String> primitiveProperties = new ArrayList<String>();

String[] names = ClassDescriptor.loadParameterNames(lc);
Class<?>[] types = lc.getParameterTypes();
assertThat(types.length, is(names.length));
for (int i=0; i<types.length; i++) {
    Object lv = ReflectionUtils.getPublicProperty(lhs, names[i]);
    Object rv = ReflectionUtils.getPublicProperty(rhs, names[i]);

    if (lv != null && rv != null && Iterable.class.isAssignableFrom(types[i])) {
        Iterable lcol = (Iterable) lv;
        Iterable rcol = (Iterable) rv;
        Iterator ltr,rtr;
        for (ltr=lcol.iterator(), rtr=rcol.iterator(); ltr.hasNext() && rtr.hasNext();){
            Object litem = ltr.next();
            Object ritem = rtr.next();

            if (findDataBoundConstructor(litem.getClass())!=null) {
                assertEqualsDataBoundBeans(litem,ritem);
            } else {
                assertThat(ritem, is(litem));
            }
        }
        assertThat("collection size mismatch between " + lhs + " and " + rhs, ltr.hasNext()
            is(false));
    } else
    if (findDataBoundConstructor(types[i])!=null || (lv!=null && findDataBoundConstructo
        // recurse into nested databound objects
        assertEqualsDataBoundBeans(lv,rv);
    } else {
        primitiveProperties.add(names[i]);
    }
}

// compare shallow primitive properties
if (!primitiveProperties.isEmpty())
    assertEqualsBeans(lhs,rhs,Util.join(primitiveProperties,""));

*
Makes sure that two collections are identical via {@link #assertEqualDataBoundBeans(Object
/
public void assertEqualsDataBoundBeans(List<?> lhs, List<?> rhs) throws Exception {
    assertEquals(lhs.size(), rhs.size());
}
    
```

```

54     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, polLocation.County)
55     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, null)
56 })
57 tierQuery.or(\orCriteria -> {
58     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, polLocation.City)
59     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, null)
60 })
61 tierQuery.or(\orCriteria -> {
62     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, polLocation.PostalCode)
63     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, null)
64 })
65
66 var results = tierQuery.select()
67 if(results.HasElements) {
68     var result = results.firstWhere( \ elt -> elt.ZipCode == polLocation.PostalCode and
69         elt.County == polLocation.County and elt.City == polLocation.City)
70     var result2 = results.firstWhere( \ elt -> elt.County == polLocation.County and elt.City == polLocation.City)
71     var result3 = results.firstWhere( \ elt -> elt.County == polLocation.County)
72
73     if(result != null) {

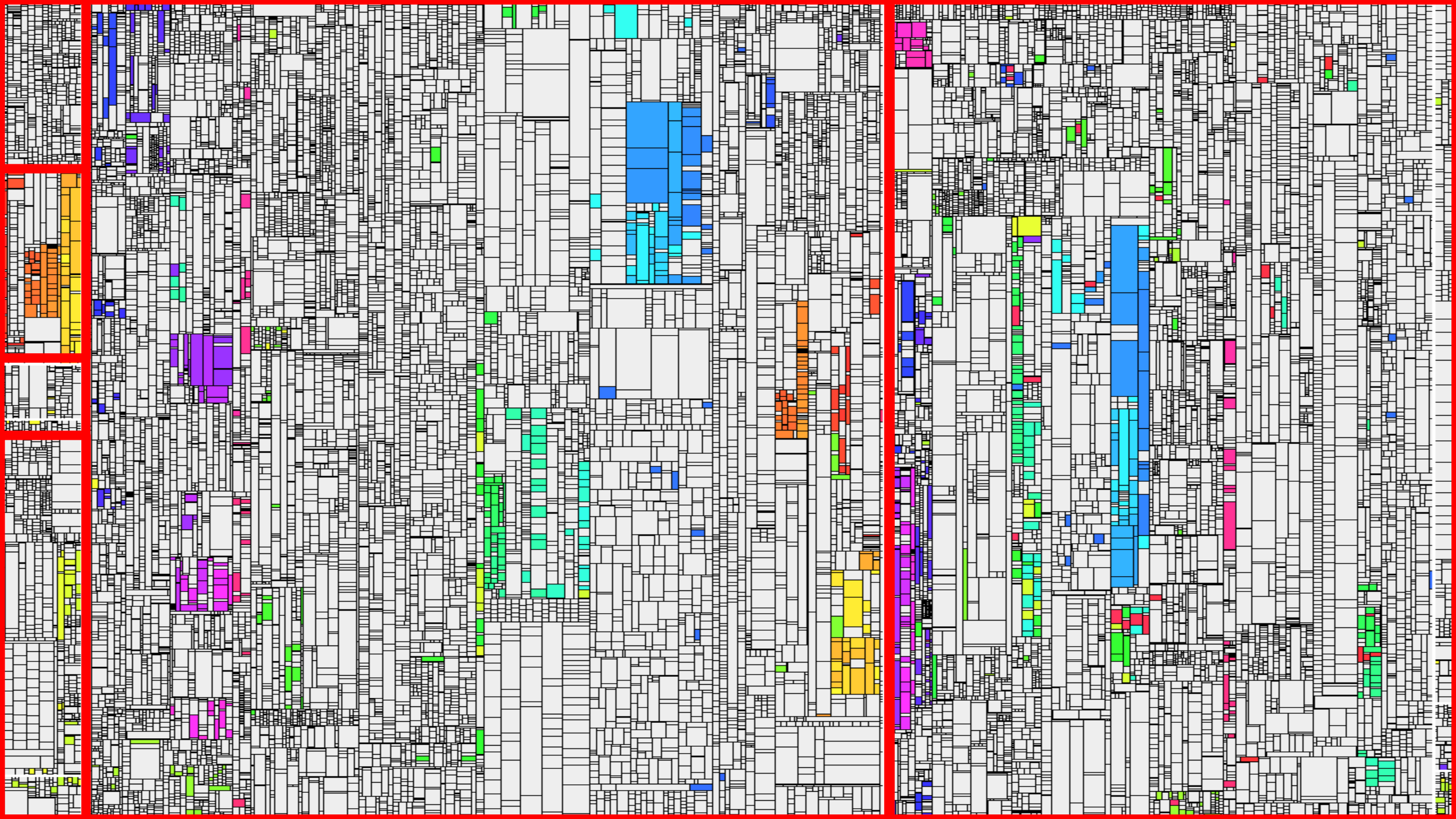
```

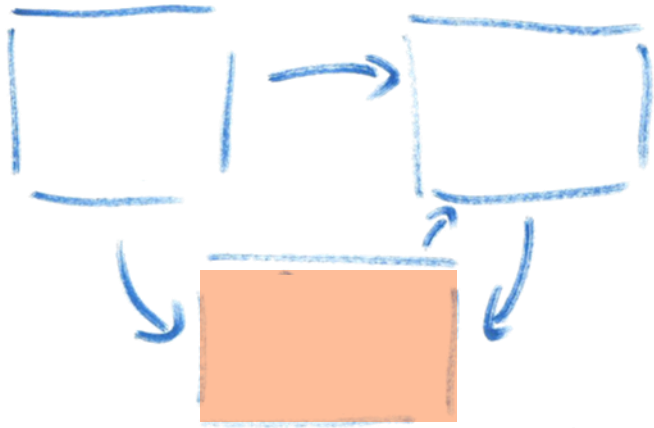
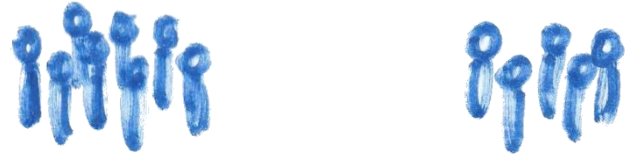
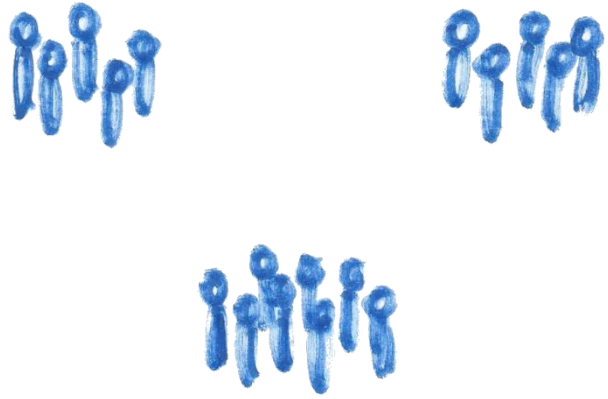
```

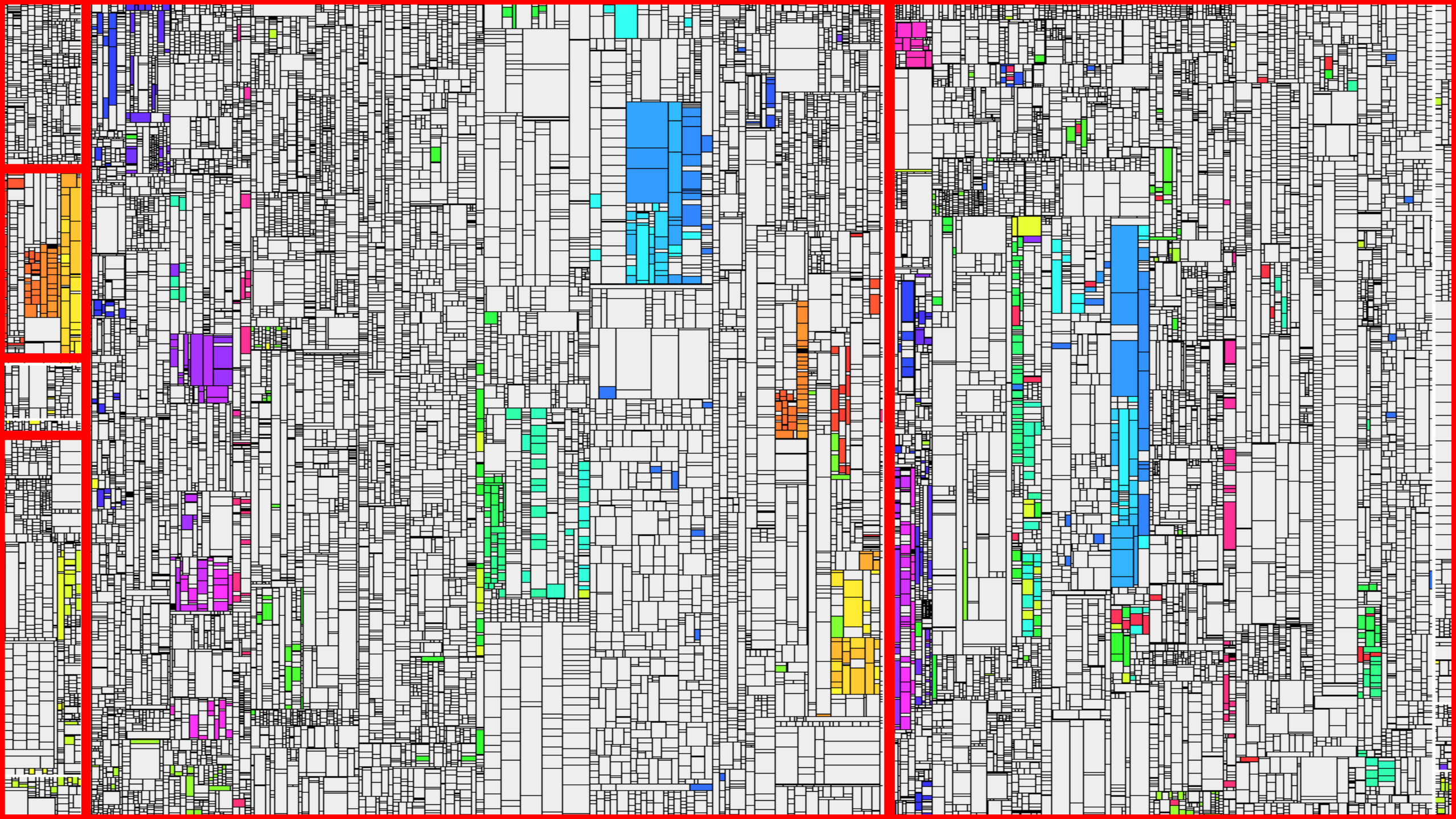
54     orCriteria.compareIgnoreCase(TerrorismTargetTier_Ext#County, Equals, polLocation.County)
55     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, null)
56 })
57 tierQuery.or(\orCriteria -> {
58     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, polLocation.City)
59     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, null)
60 })
61 tierQuery.or(\orCriteria -> {
62     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, polLocation.PostalCode)
63     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, null)
64 })
65
66 var results = tierQuery.select()
67 if(results.HasElements) {
68     var result = results.firstWhere( \ elt -> elt.ZipCode == polLocation.PostalCode and
69         elt.County == polLocation.County and elt.City == polLocation.City)
70     var result2 = results.firstWhere( \ elt -> elt.County == polLocation.County and elt.City == polLocation.City)
71     // - Defect 224 - Fix for defaulting target tier value irrespective of lower and upper
72     var result3 = results.firstWhere( \ elt -> elt.County.equalsIgnoreCase(polLocation.County))
73
74     if(result != null) {

```


Was tun?







```

54     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, polLocation.County)
55     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, null)
56 })
57 tierQuery.or(\orCriteria -> {
58     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, polLocation.City)
59     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, null)
60 })
61 tierQuery.or(\orCriteria -> {
62     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, polLocation.PostalCode)
63     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, null)
64 })
65
66 var results = tierQuery.select()
67 if(results.HasElements) {
68     var result = results.firstWhere( \ elt -> elt.ZipCode == polLocation.PostalCode and
69         elt.County == polLocation.County and elt.City == polLocation.City)
70     var result2 = results.firstWhere( \ elt -> elt.County == polLocation.County and elt.City == polLocation.City)
71     var result3 = results.firstWhere( \ elt -> elt.County == polLocation.County)
72
73     if(result != null) {

```

```

54     orCriteria.compareIgnoreCase(TerrorismTargetTier_Ext#County, Equals, polLocation.County)
55     orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, null)
56 })
57 tierQuery.or(\orCriteria -> {
58     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, polLocation.City)
59     orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, null)
60 })
61 tierQuery.or(\orCriteria -> {
62     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, polLocation.PostalCode)
63     orCriteria.compare(TerrorismTargetTier_Ext#ZipCode, Equals, null)
64 })
65
66 var results = tierQuery.select()
67 if(results.HasElements) {
68     var result = results.firstWhere( \ elt -> elt.ZipCode == polLocation.PostalCode and
69         elt.County == polLocation.County and elt.City == polLocation.City)
70     var result2 = results.firstWhere( \ elt -> elt.County == polLocation.County and elt.City == polLocation.City)
71     // - Defect 224 - Fix for defaulting target tier value irrespective of lower and upper
72     var result3 = results.firstWhere( \ elt -> elt.County.equalsIgnoreCase(polLocation.County))
73
74     if(result != null) {

```



reset publicCgeoDir after dir permissions granted (fix #7890)

by moving-bits as revision 40e4dc51 in master (cgeo-git)

Oct 13 2019
11:06

⌵ Affected files 3

🔔 1 Alerts

Message

Context

Found potential inconsistent clone change in PermissionHandler.java

Broken clone

Old clone finding

Code change

Produkte einer Produktfamilie

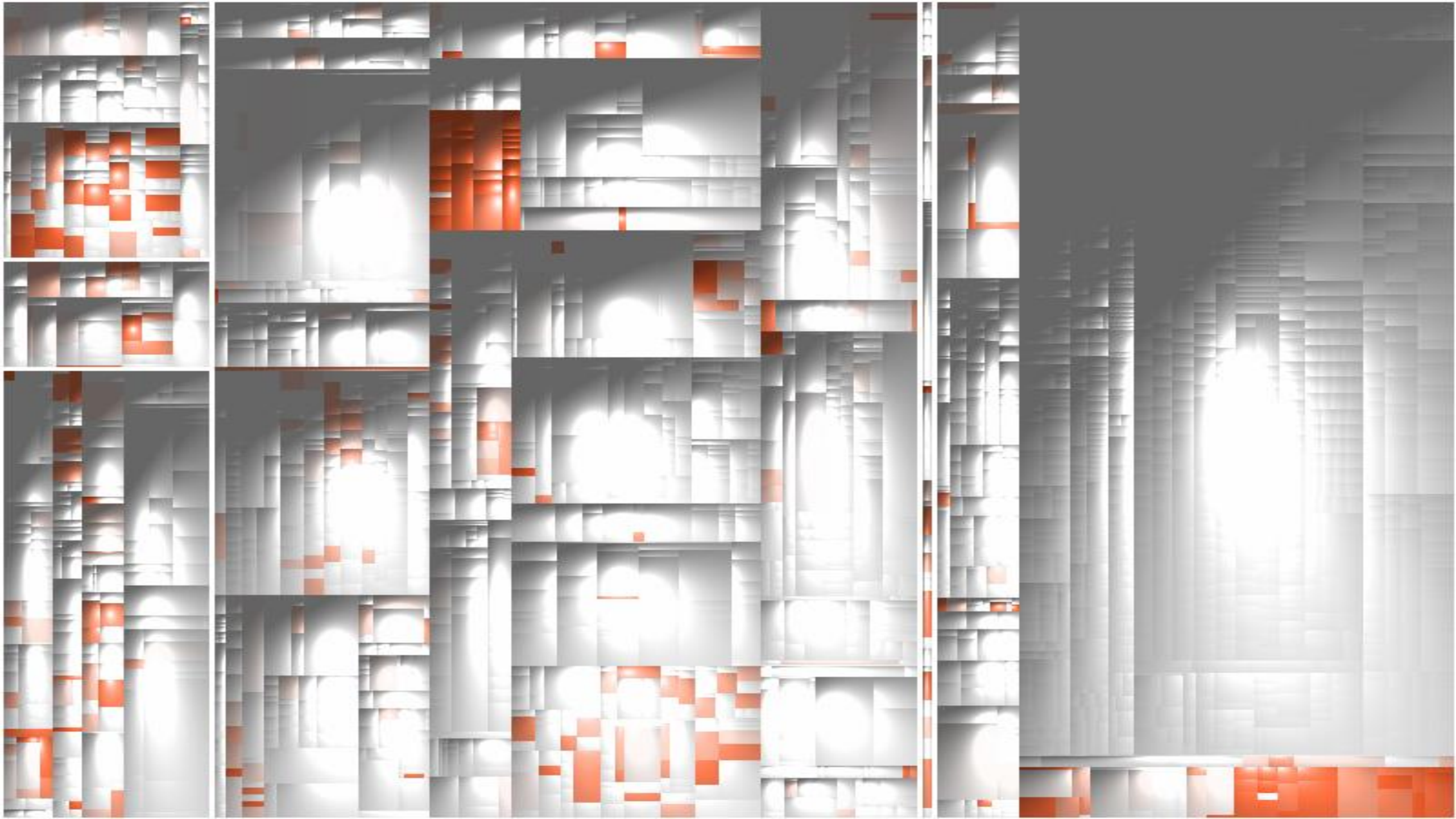
Product 1

Product 2

Product 3

Product 4

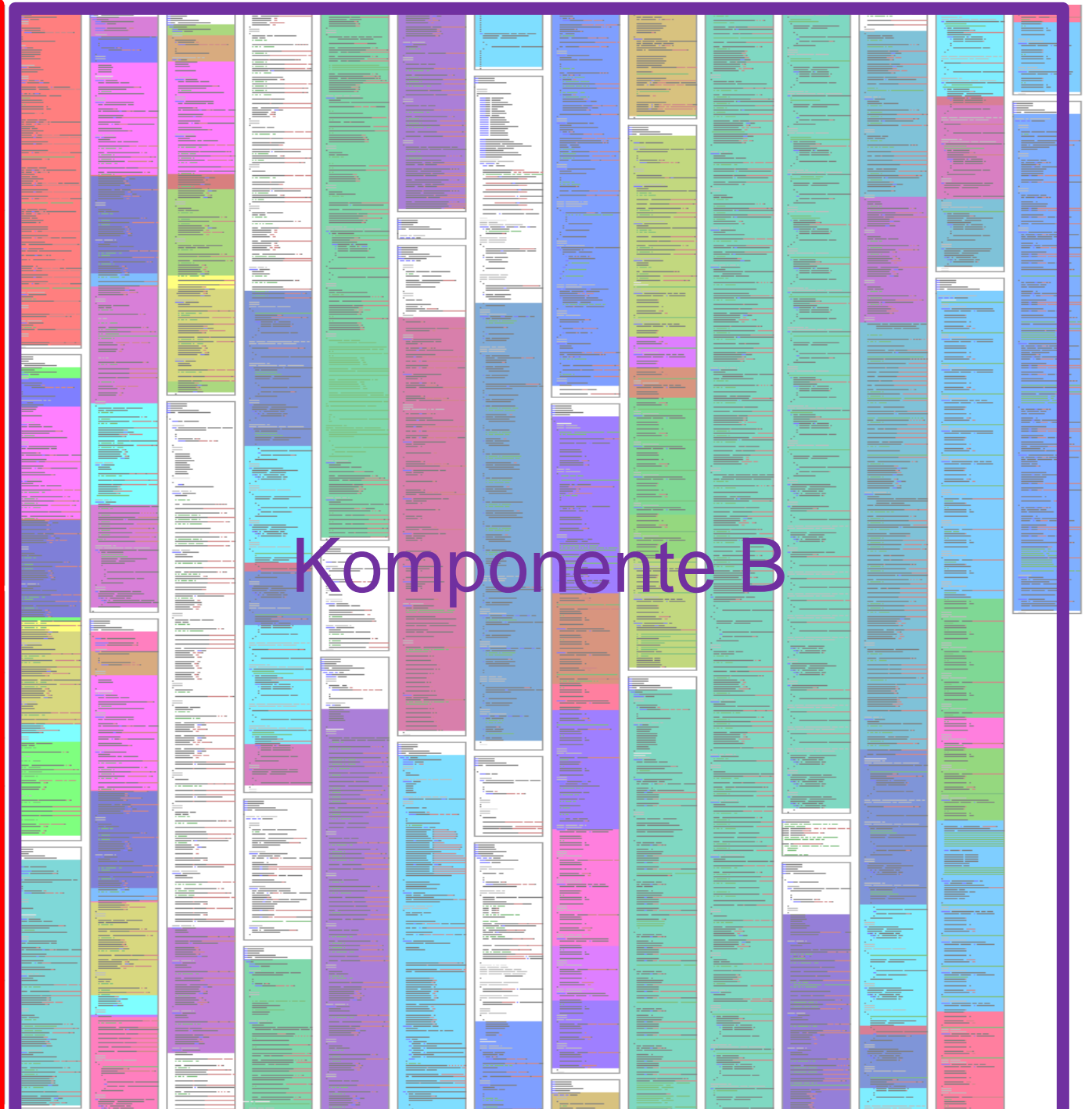
Product 5



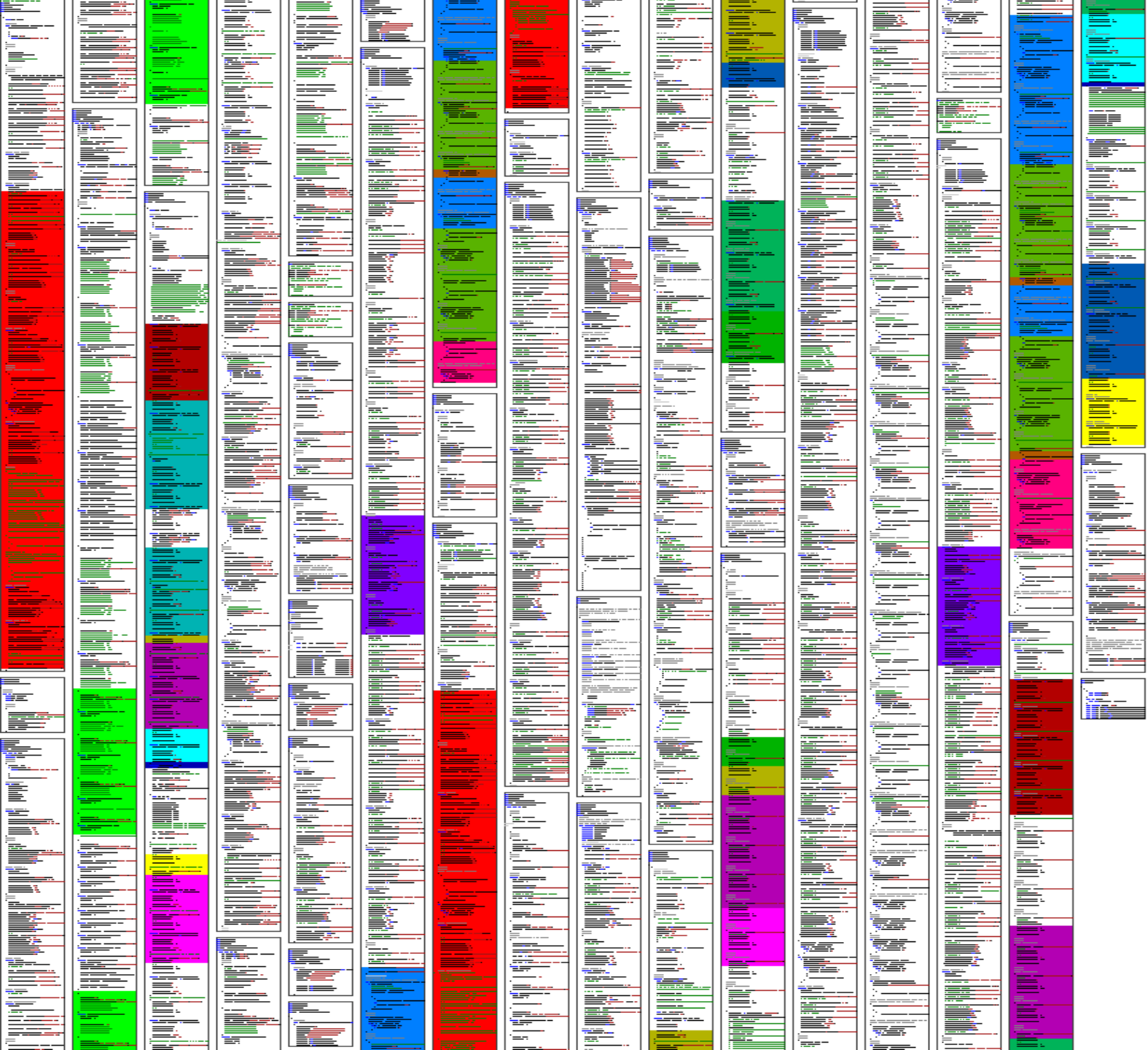




Komponente A

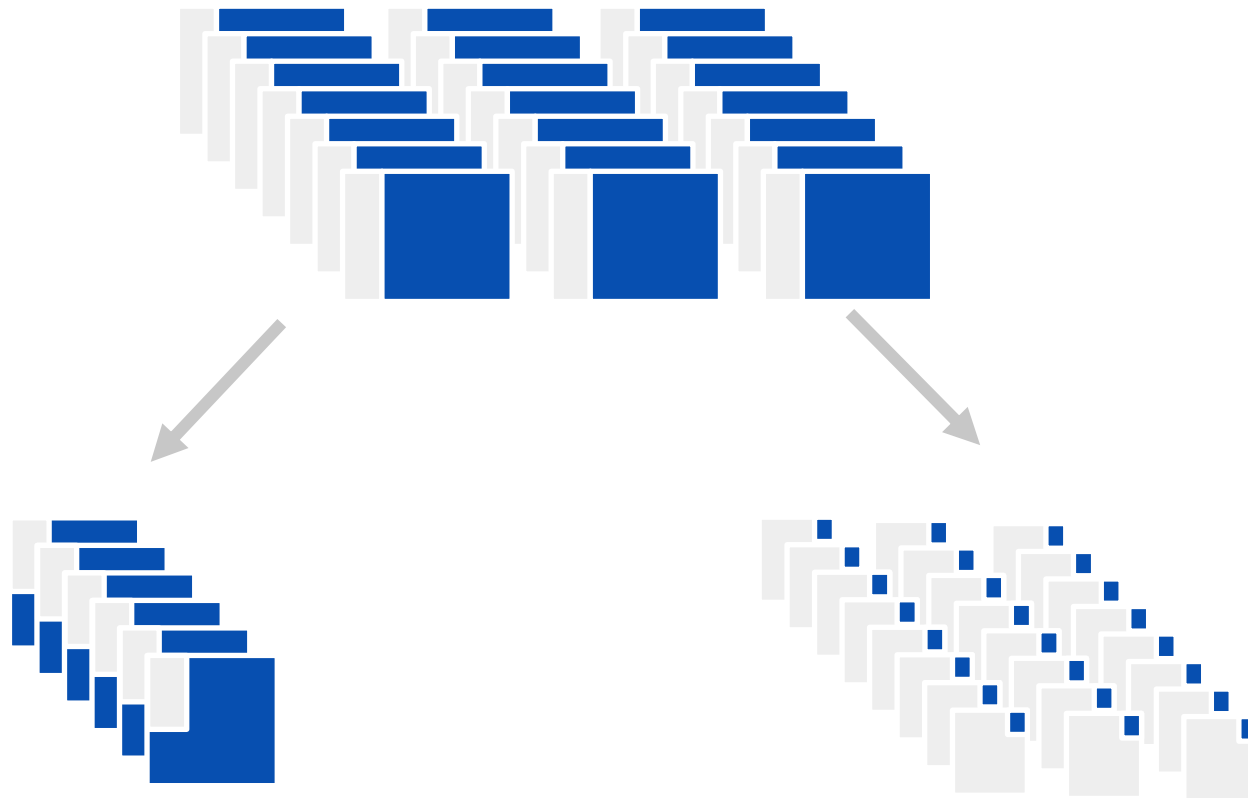


Komponente B

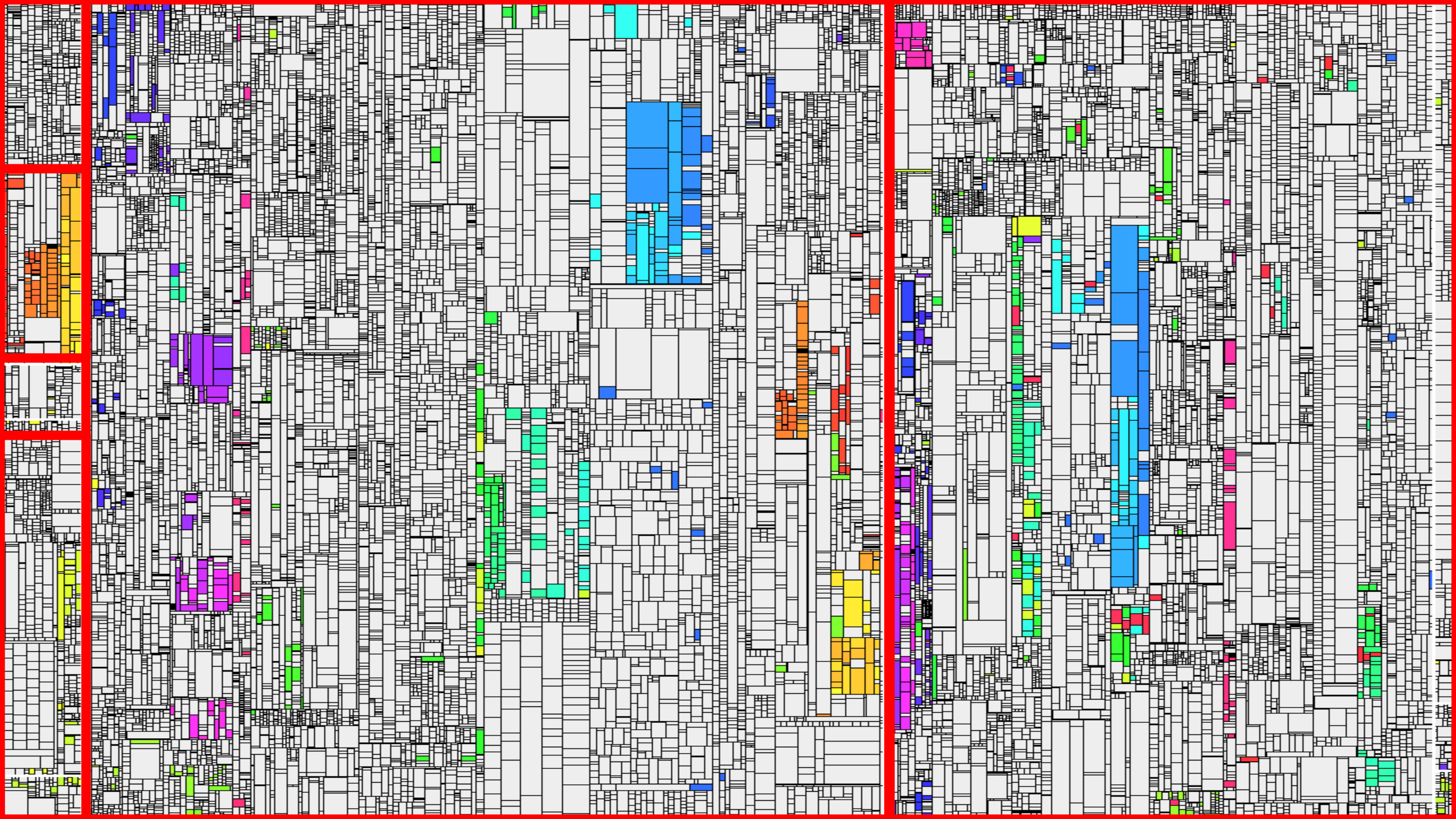


Buildsystem einer Anwendungslandschaft

Ad-Hoc Reuse in einem Build-System



	Ad hoc Reuse	Systematisch
Build Definitions	81	94
Lines of Code	457.388	91.734
Clone Coverage	97,5%	82,8%
Blow-Up	929%	194%



Kontakt – Ich freue mich auf Diskussionen 😊



Dr. Elmar Jürgens · juergens@cqse.eu · +49 179 675 3863

CQSE GmbH
Lichtenbergstraße 8
85748 Garching bei München
www.cqse.eu

