

Systematic Reuse or Copy&Paste?

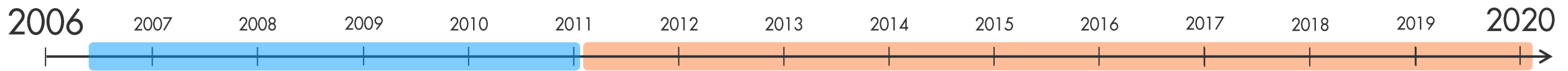
Experiences from 10 years of Software-Quality-Analysis in Research & Practice



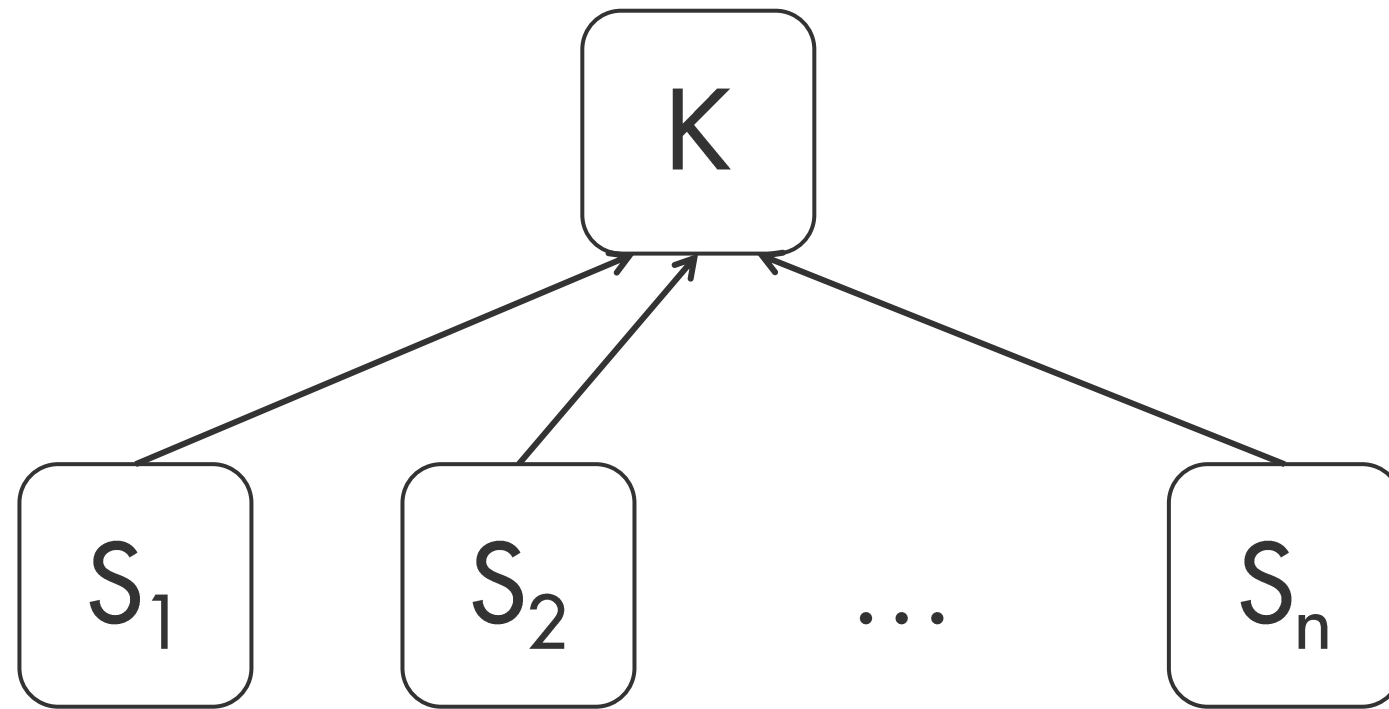
TUM



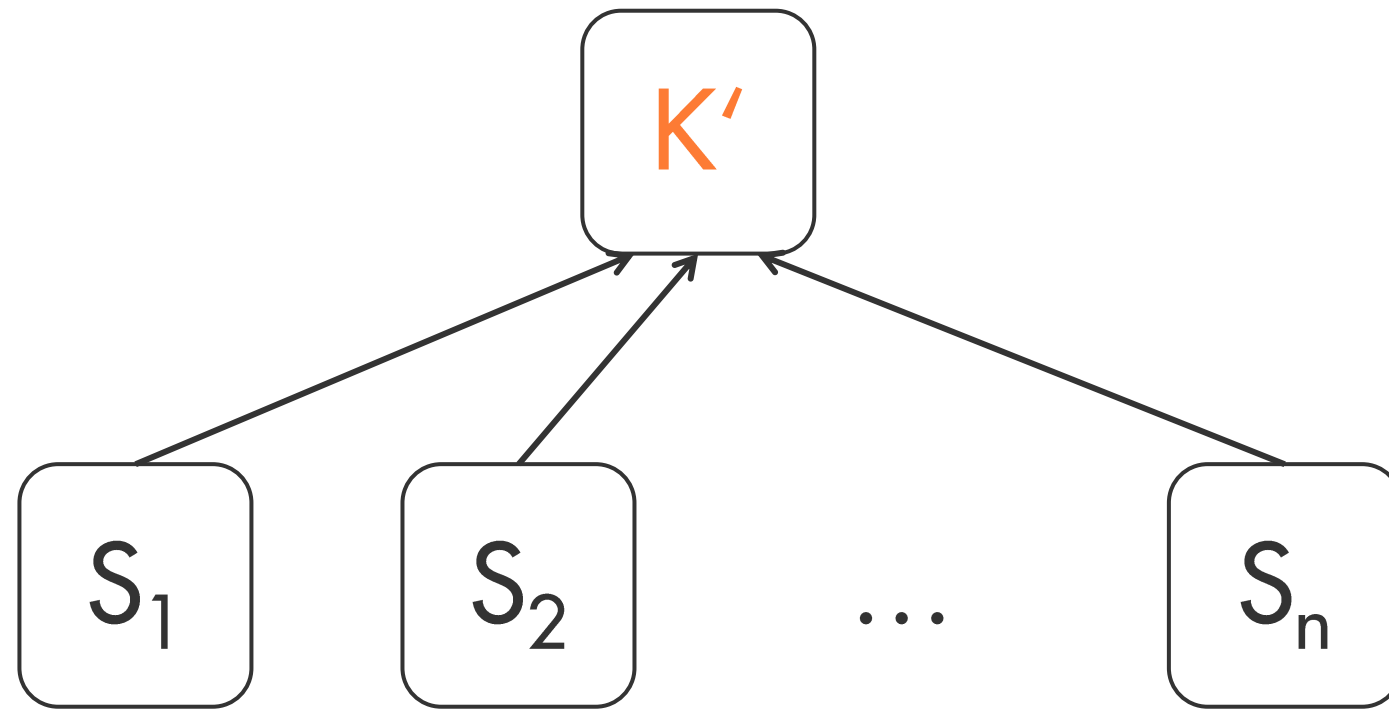
CQSE



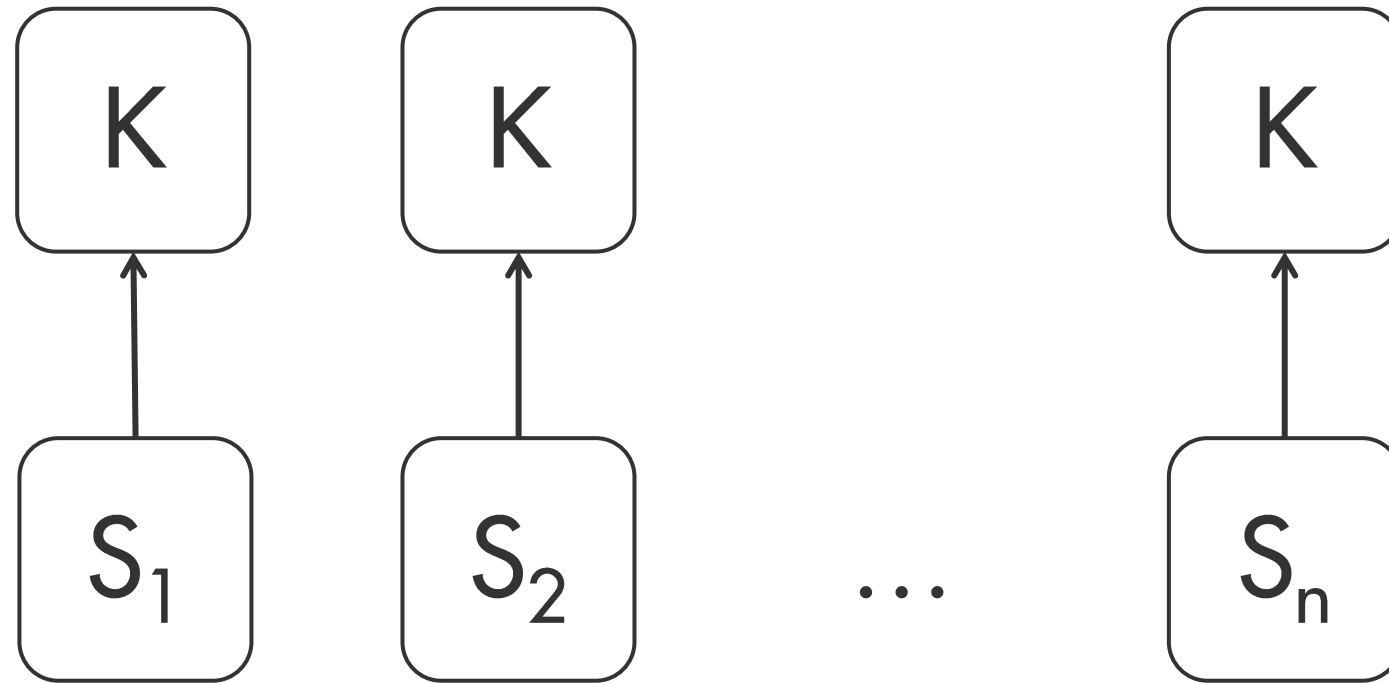
Systematic Reuse



Systematic 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();
}

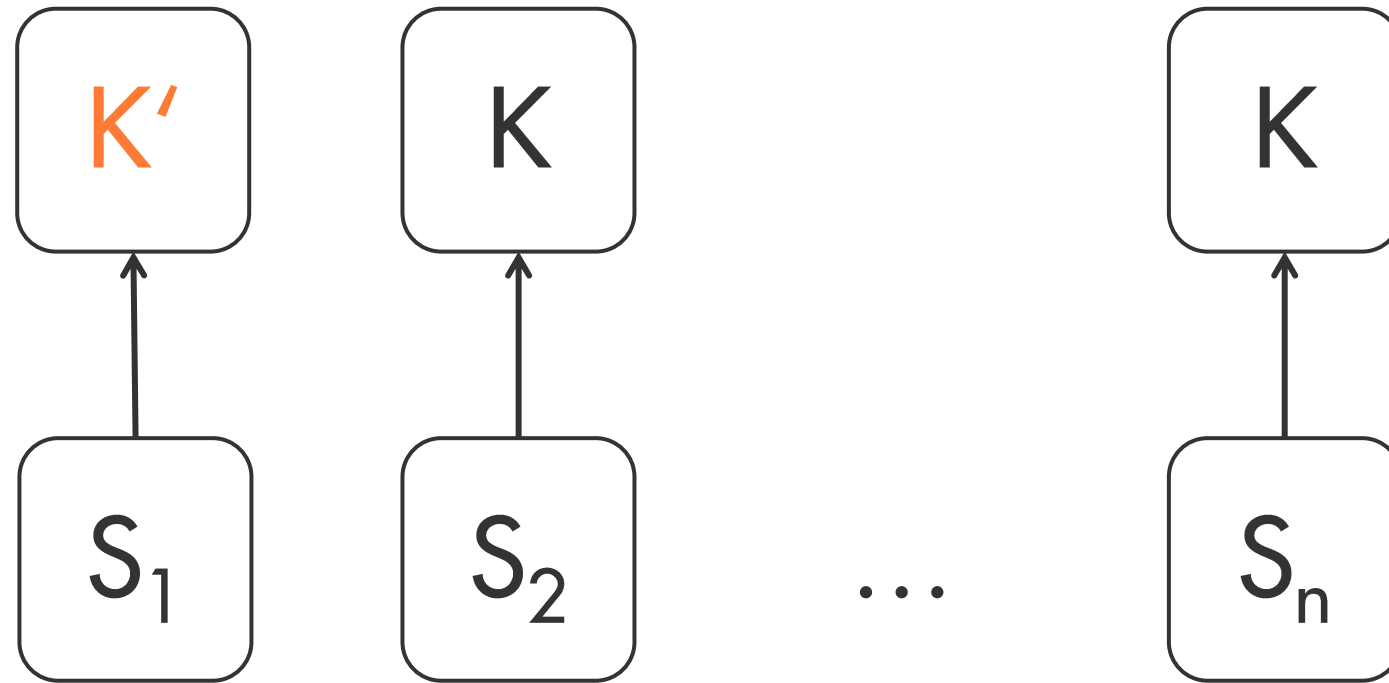
```

```

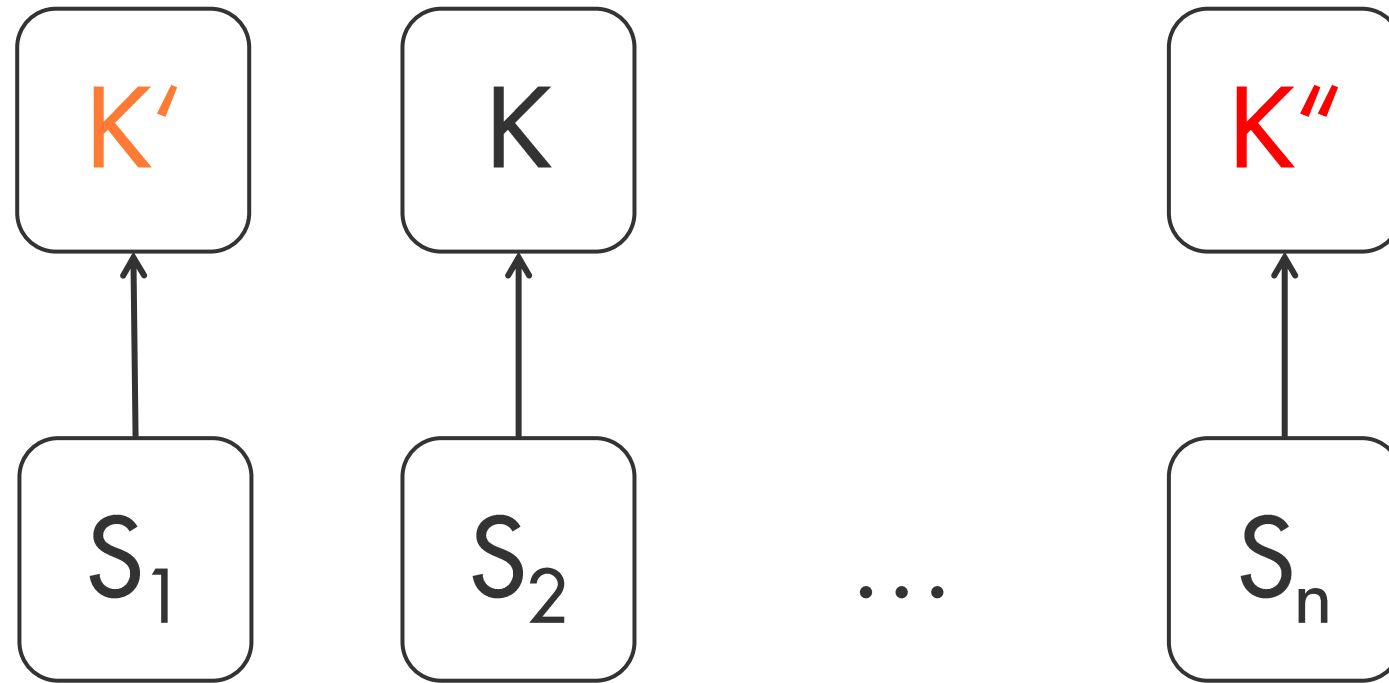
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        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();
}
```

**How much Ad-Hoc Reuse
do we find in practice?**

```

// 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();
}

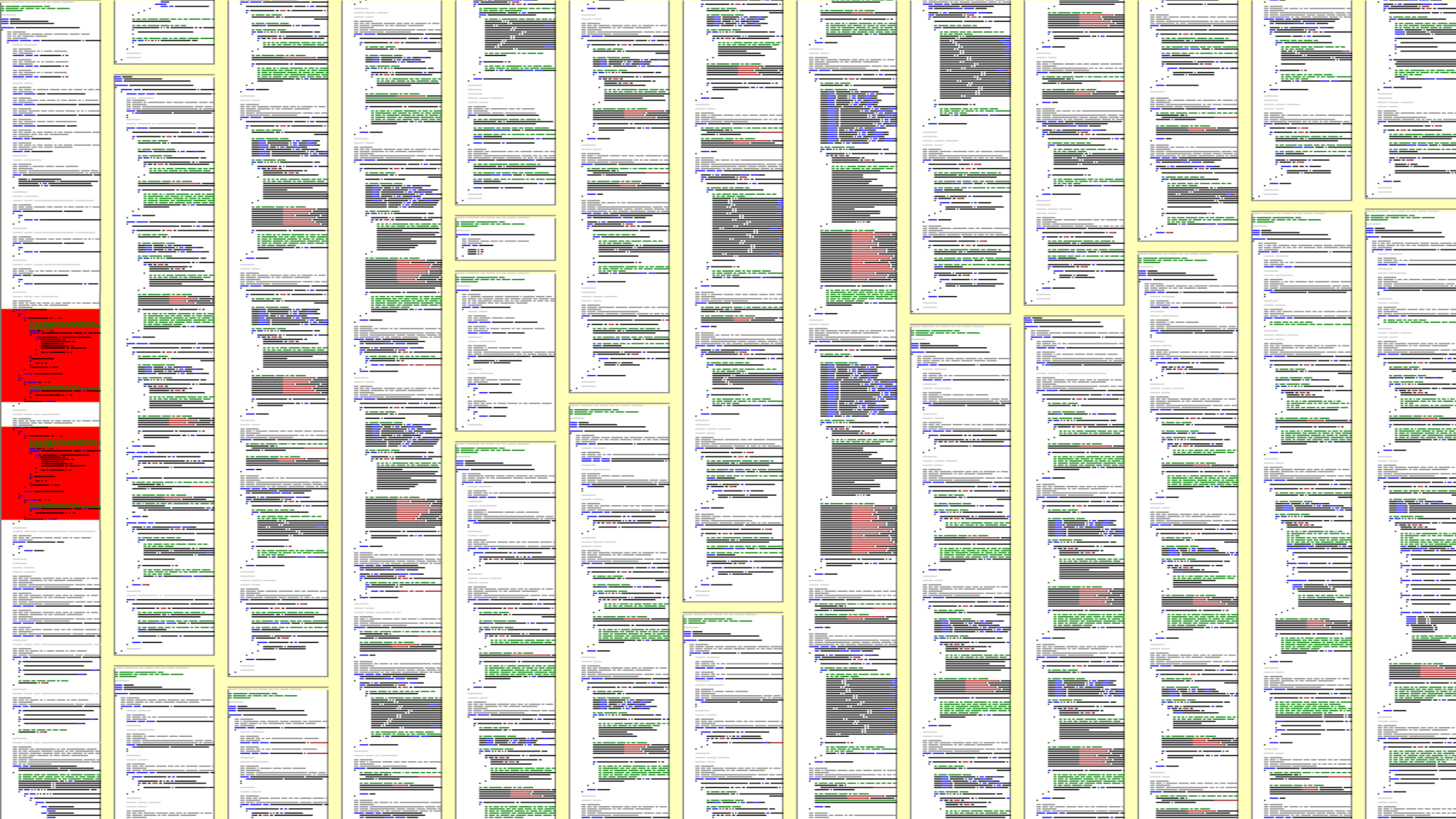
```

```

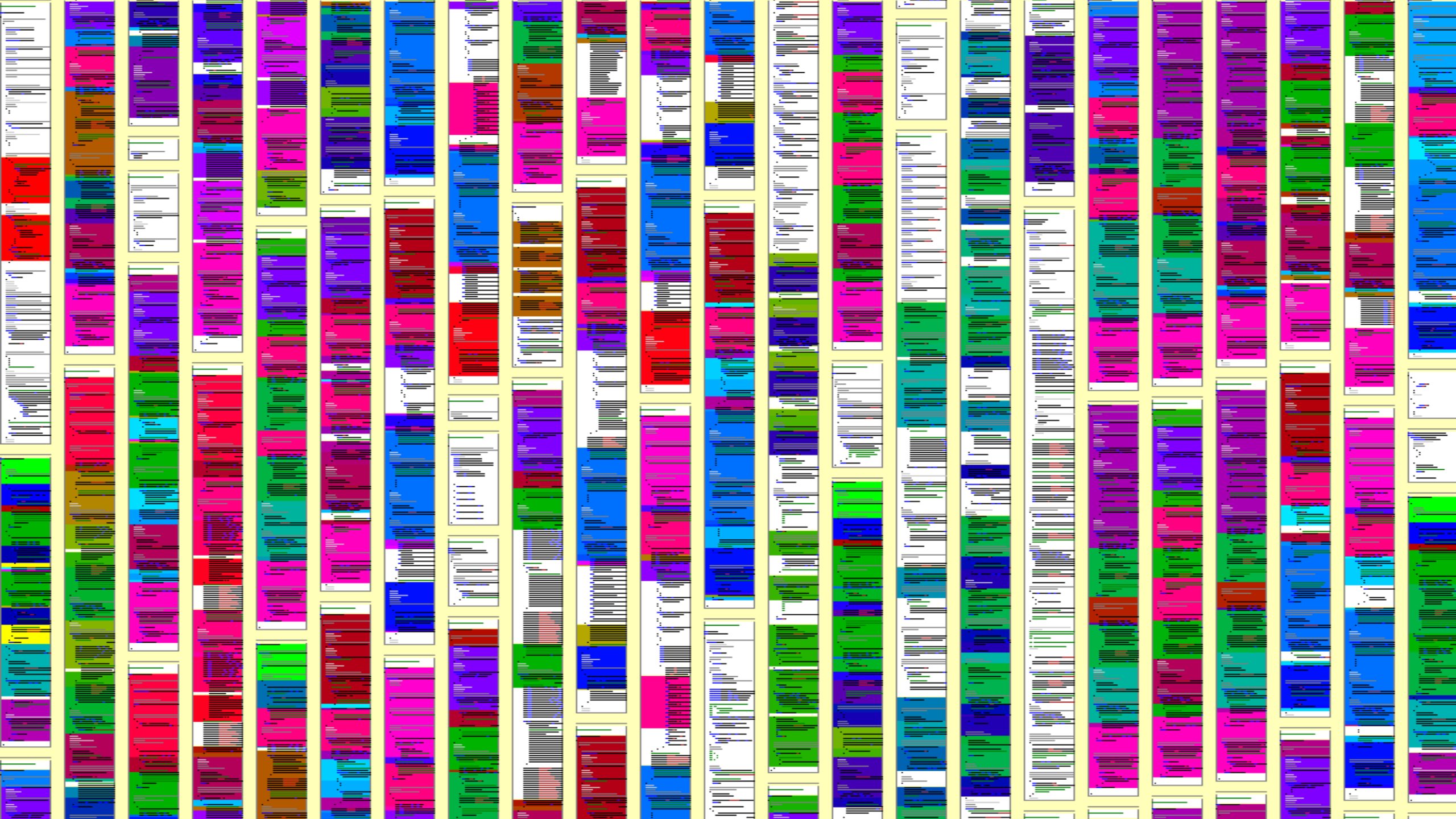
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        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();
}

```









**Does this cause
problems?**


```
// 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();
}
```

Study

Munich Re 

- Over 100 Bugs in productive Software



- 52% of all unintentional inconsistencies are buggy

Juergens, Deissenboeck et al: *Do Code Clones Matter?* ICSE 2009

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 && findDataBoundConstructor
        // 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 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());
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 && findDataBoundConstructor
        // 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());
}

```

```

/* Process the input string received prior to the
newline. */
do
{
    /* Pass the string to FreeRTOS+CLI. */
    xMoreDataToFollow = FreeRTOS_CLIProcessCommand( cInputString, cOutputString );

    /* Send the output generated by the command's
implementation. */
    sendto( xSocket, cOutputString, strlen( cOutputString ), 0, ( SOCKADDR * ) &xClient );

} while( xMoreDataToFollow != pdFALSE ); /* Until the command does not return more data. */

/* All the strings generated by the command processing
have been sent. Clear the input string ready to receive
the next command. */
cInputIndex = 0;
memset( cInputString, 0x00, cmdMAX_INPUT_SIZE );

/* Transmit a spacer, just to make the command console
easier to read. */
sendto( xSocket, "\r\n", strlen( "\r\n" ), 0, ( SOCKADDR * ) &xClient );
}
else
{
    if( cInChar == '\r' )
    {
        /* Ignore the character. Newlines are used to
detect the end of the input string. */
    }
    else if( cInChar == '\b' )
    {
        /* Backspace was pressed. Erase the last character
in the string - if any. */
        if( cInputIndex > 0 )
        {
            cInputIndex--;
            cInputString[ cInputIndex ] = '\0';
        }
    }
    else
    {

```

```

/* Process the input string received prior to the
newline. */
do
{
    /* Pass the string to FreeRTOS+CLI. */
    xMoreDataToFollow = FreeRTOS_CLIProcessCommand( cInputString, cOutputString );

    /* Send the output generated by the command's
implementation. */
    sendto( xSocket, cOutputString, strlen( cOutputString ), 0, ( SOCKADDR * ) &xClient );

} while( xMoreDataToFollow != pdFALSE ); /* Until the command does not return more data. */

/* All the strings generated by the command processing
have been sent. Clear the input string ready to receive
the next command. */
cInputIndex = 0;
memset( cInputString, 0x00, cmdMAX_INPUT_SIZE );

/* Transmit a spacer, just to make the command console
easier to read. */
sendto( xSocket, "\r\n", strlen( "\r\n" ), 0, ( SOCKADDR * ) &xClient );
}
else
{
    if( cInChar == '\r' )
    {
        /* Ignore the character. Newlines are used to
detect the end of the input string. */
    }
    else if( ( cInChar == '\b' ) || ( cInChar == cmdASCII_DEL ) )
    {
        /* Backspace was pressed. Erase the last character
in the string - if any. */
        if( cInputIndex > 0 )
        {
            cInputIndex--;
            cInputString[ cInputIndex ] = '\0';
        }
    }
    else
    {

```

```

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 = 'I'.
    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 = 'I'.
    d_fibu_type = text-k01.
  .

```

larger/source/.../PriceElementUtil.plsql

(revision default:1...)

```

223     AND NVL(t.valid_from, Database_SYS.Get_First_C
224     AND NVL(t.valid_to, Database_SYS.Get_Last_Cale
225
226 BEGIN
227     OPEN get_price_interval;
228     FETCH get_price_interval INTO valid_from_, valid_u
229     CLOSE get_price_interval;
230     IF valid_from_ > Site_API.Get_Site_Date(contract_)
231         valid_date_ := valid_from_;
232     ELSE
233         valid_date_ := LEAST(NVL(valid_until_, TRUNC(Sit
234     END IF;
235
236     -- only calculate unit base (False) but no metal su
237     temp1_ :=0;
238
239     OPEN get_pl_currency_code;
240     FETCH get_pl_currency_code INTO pl_currency_code_;
241     CLOSE get_pl_currency_code;
242
243     FOR rec_ IN get_all_conn_unit_base_false LOOP

```

larger/source/.../PriceElementUtil.plsql

(revision default:1...)

```

346 BEGIN
347     OPEN get_price_interval;
348     FETCH get_price_interval INTO valid_from_, valid_u
349     CLOSE get_price_interval;
350     IF valid_from_ > Site_API.Get_Site_Date(contract_)
351         valid_date_ := valid_from_;
352     ELSE
353         -- Change for Ticket #1234 (START)
354         --valid_date_ := LEAST(NVL(valid_until_, TRUNC($
355         valid_date_ := LEAST(NVL(valid_until_, Site_API.
356         -- Change for Ticket #1234 (FINISH)
357     END IF;
358
359     -- calculate only metal surcharge without unit base
360     temp3_ :=0;
361
362     OPEN get_pl_currency_code;
363     FETCH get_pl_currency_code INTO pl_currency_code_;
364     CLOSE get_pl_currency_code;
365
366     FOR rec_ IN get_all_metal_surcharge LOOP

```



```

53 tierQuery.or(\orCriteria -> {
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) {
74

```

```

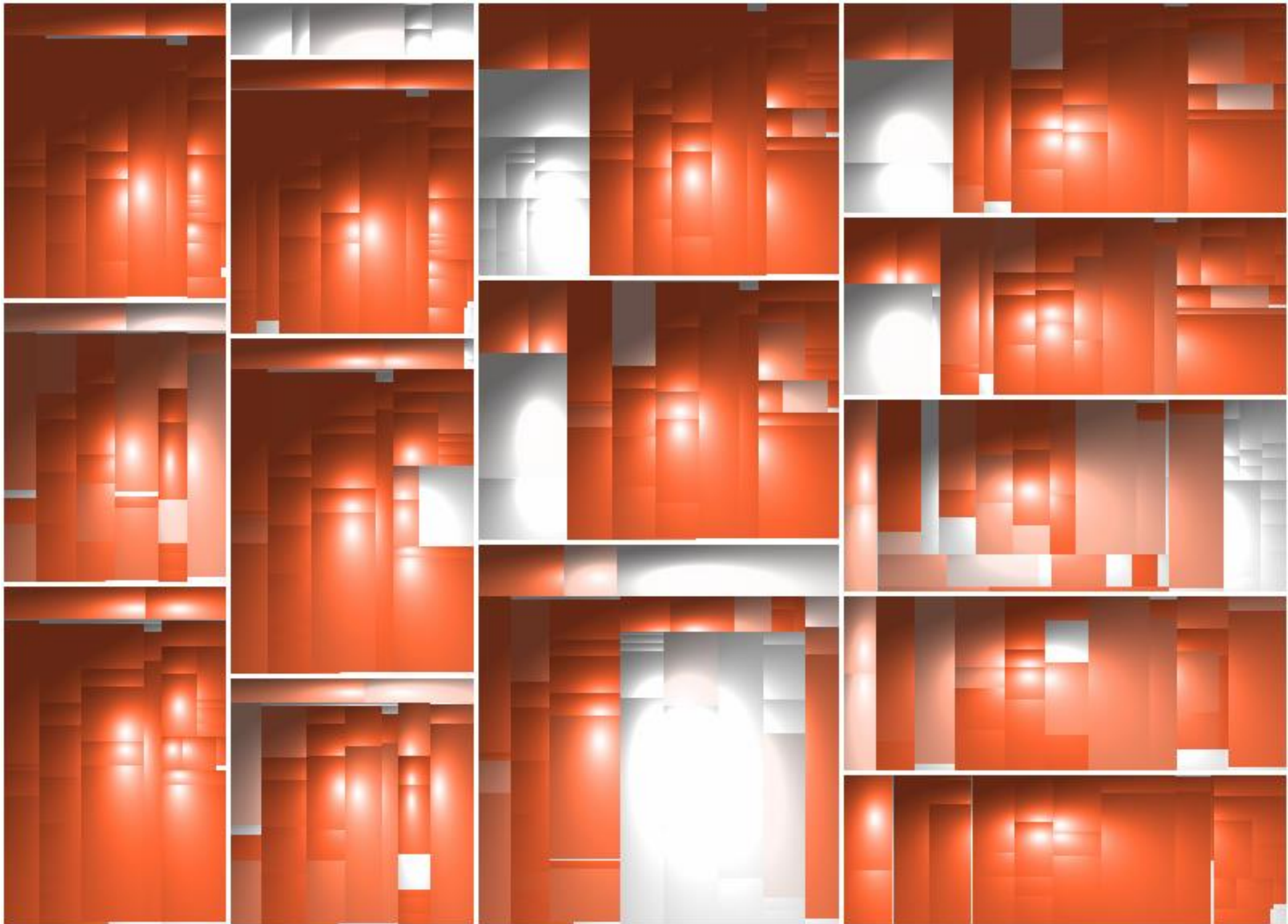
54   orCriteria.compareIgnoreCase(TerrorismTargetTier_Ext#County, Equals, polLocation.County)
55   orCriteria.compare(TerrorismTargetTier_Ext#County, Equals, null)
56 })
57 tierQuery.or(\orCriteria -> {
58   orCriteria.compare(TerrorismTargetTier_Ext#City, Equals, polLocation.City)
59   orCriteria.compare(TerrorismTargetTier_Ext#City, 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) {

```

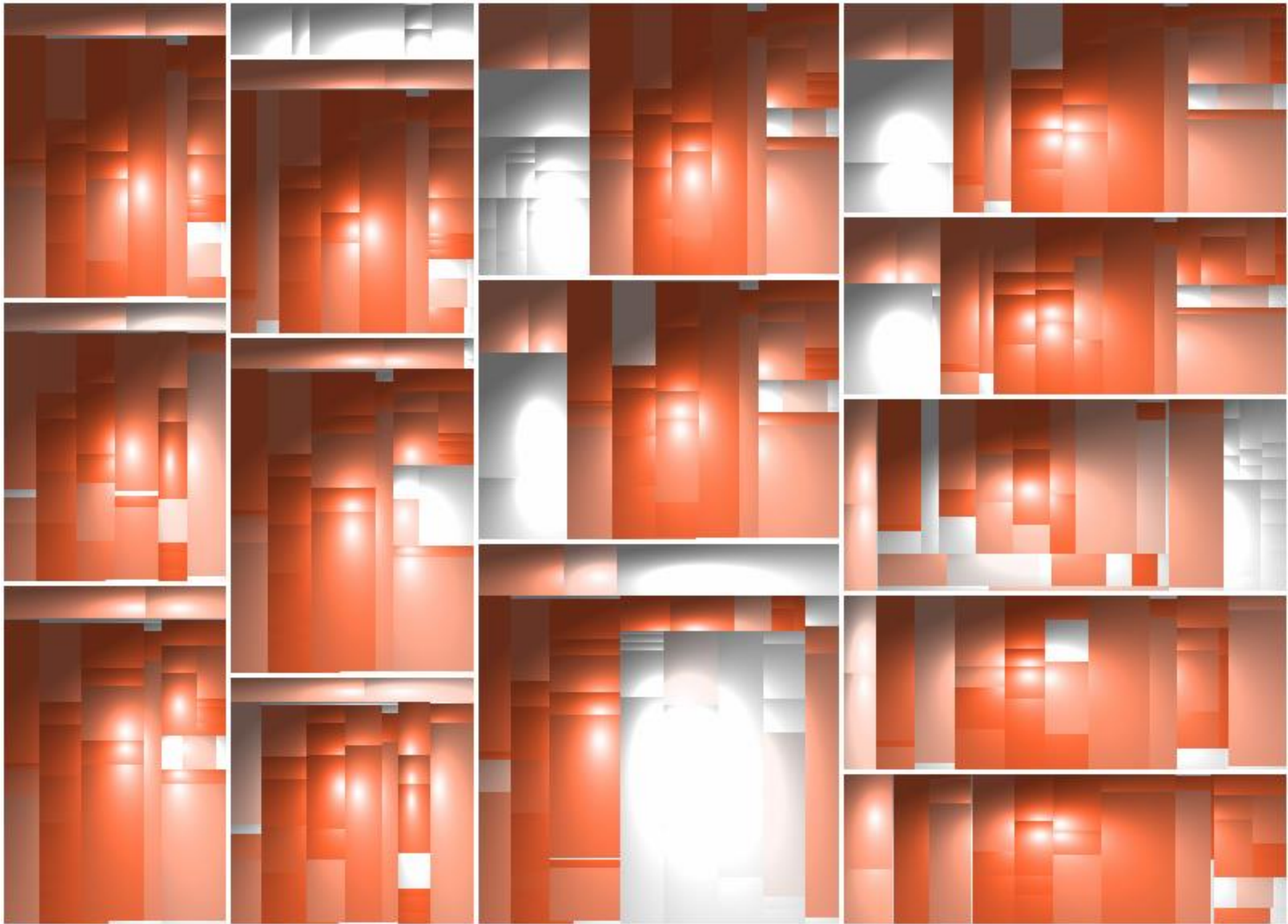
**Are there only clones
inside a single code base?**

**Products of one product
family**

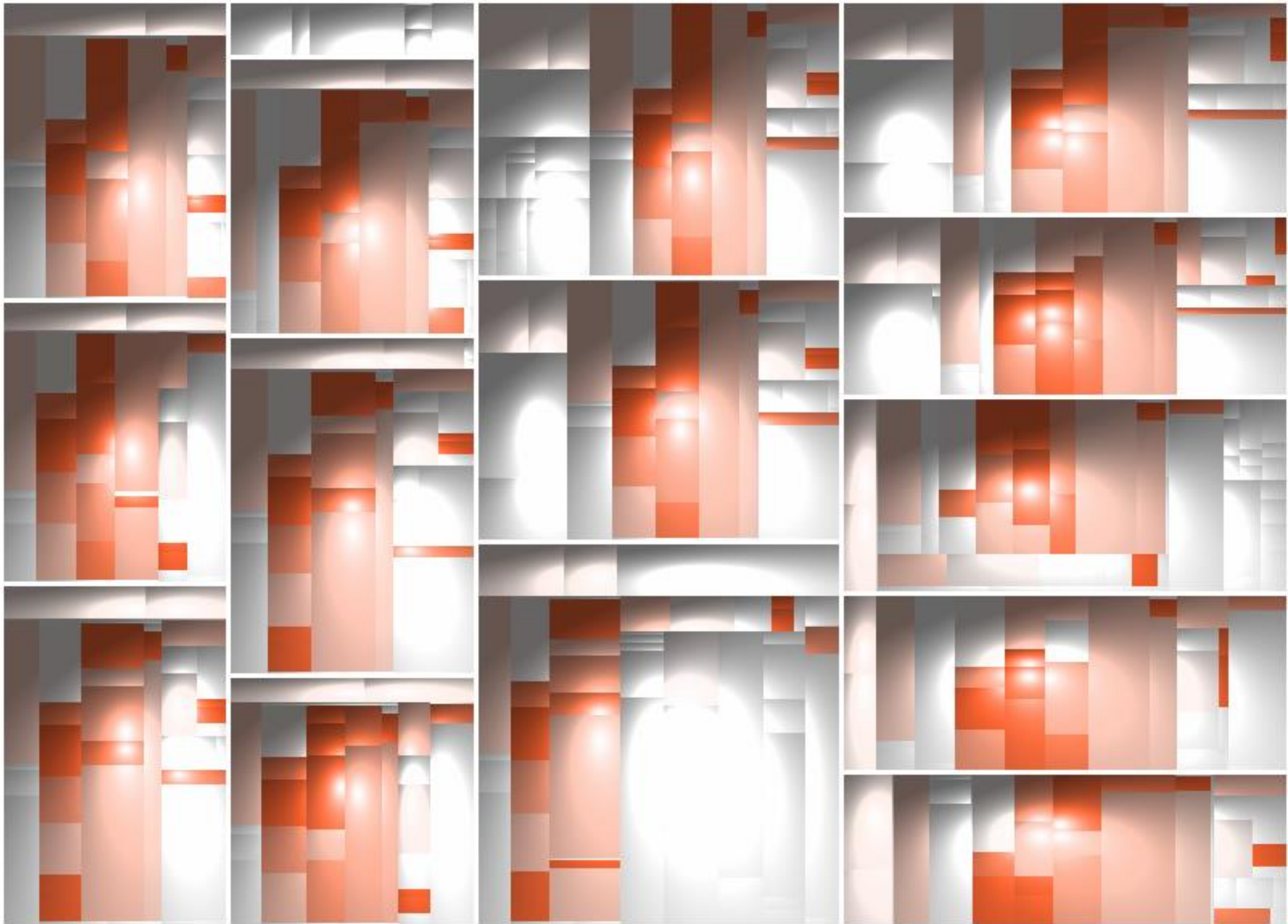




5 Projects



10 Projects



14 Projects

Products of one product family II

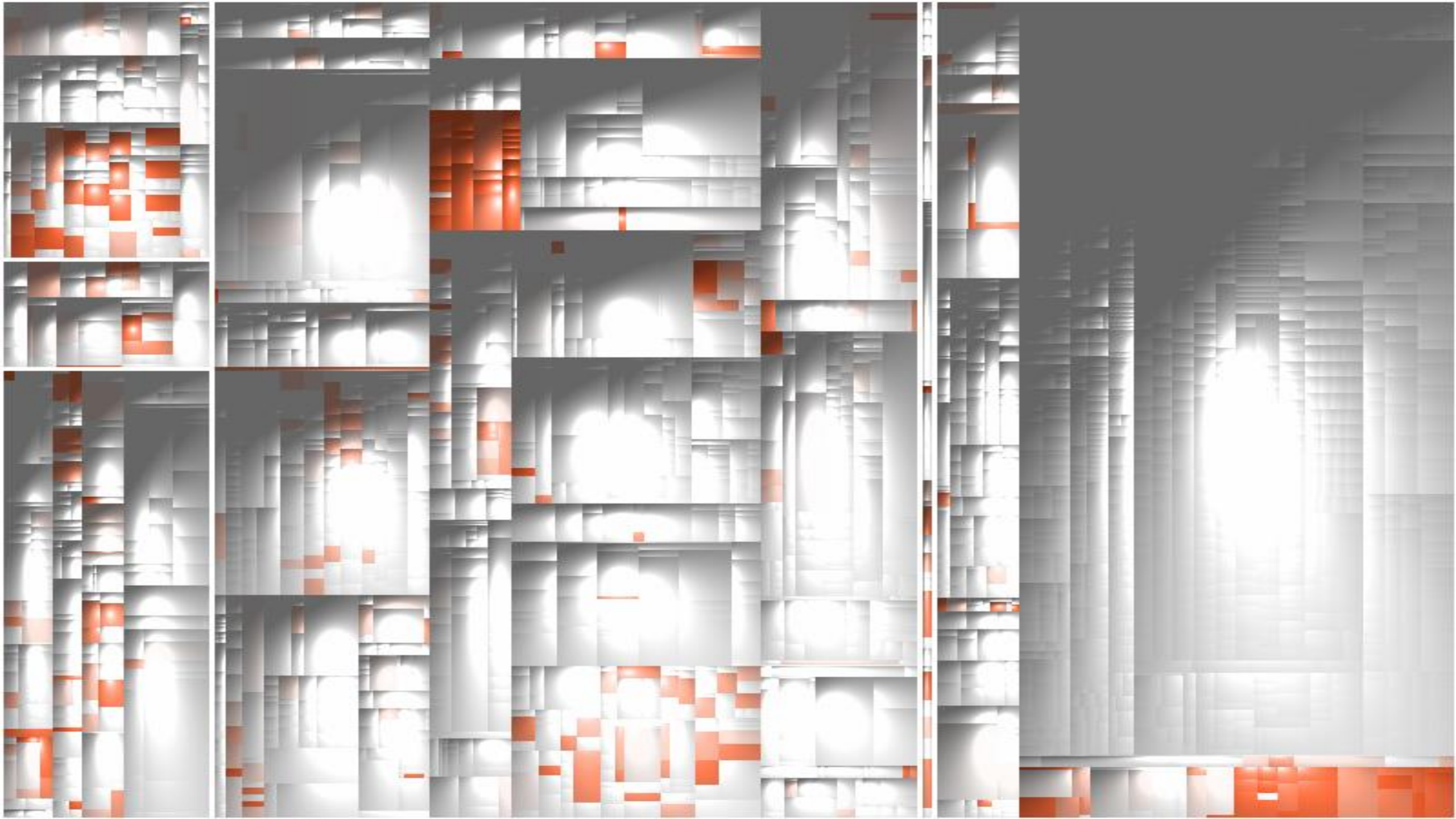
Product 1

Product 2

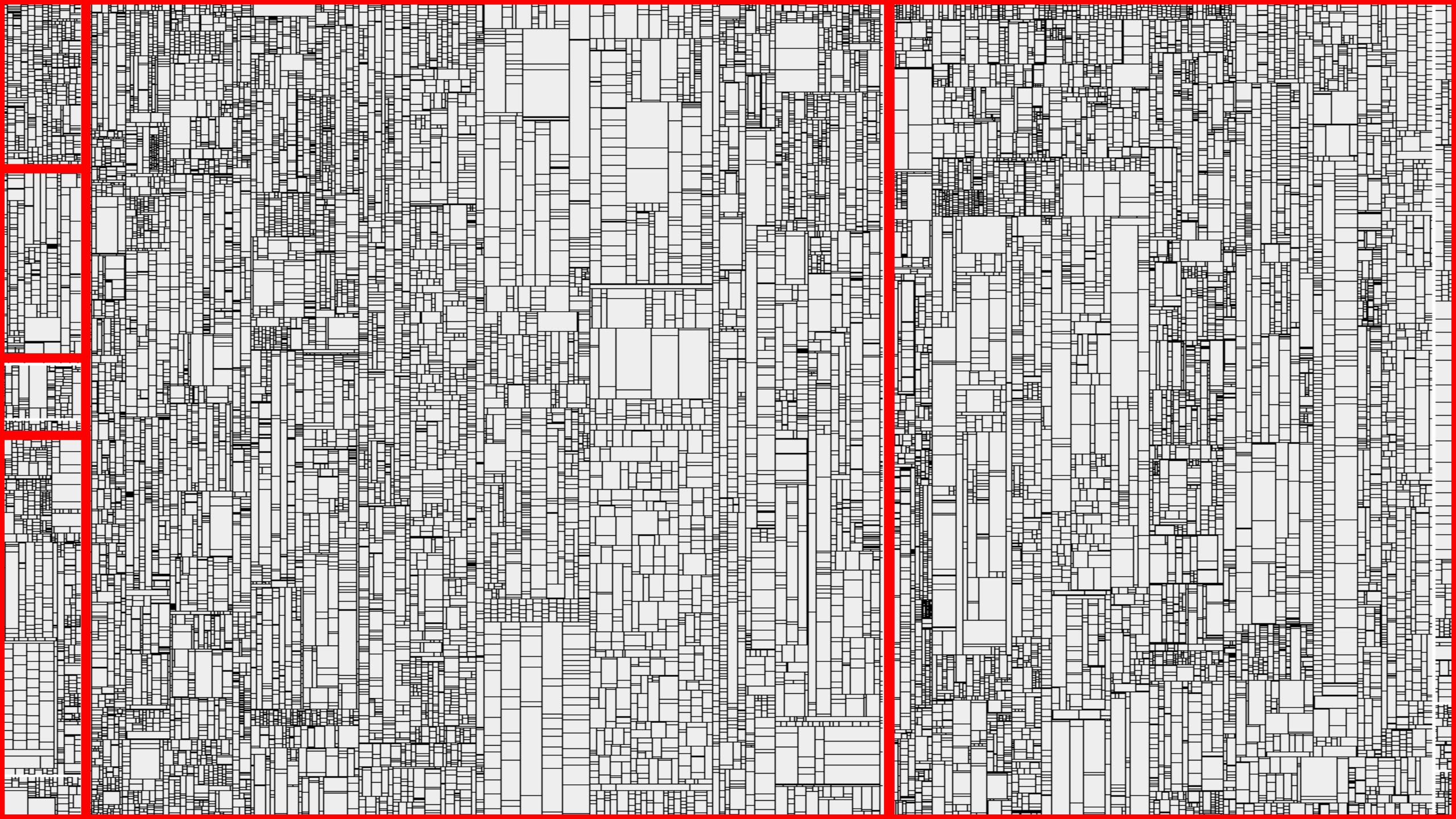
Product 3

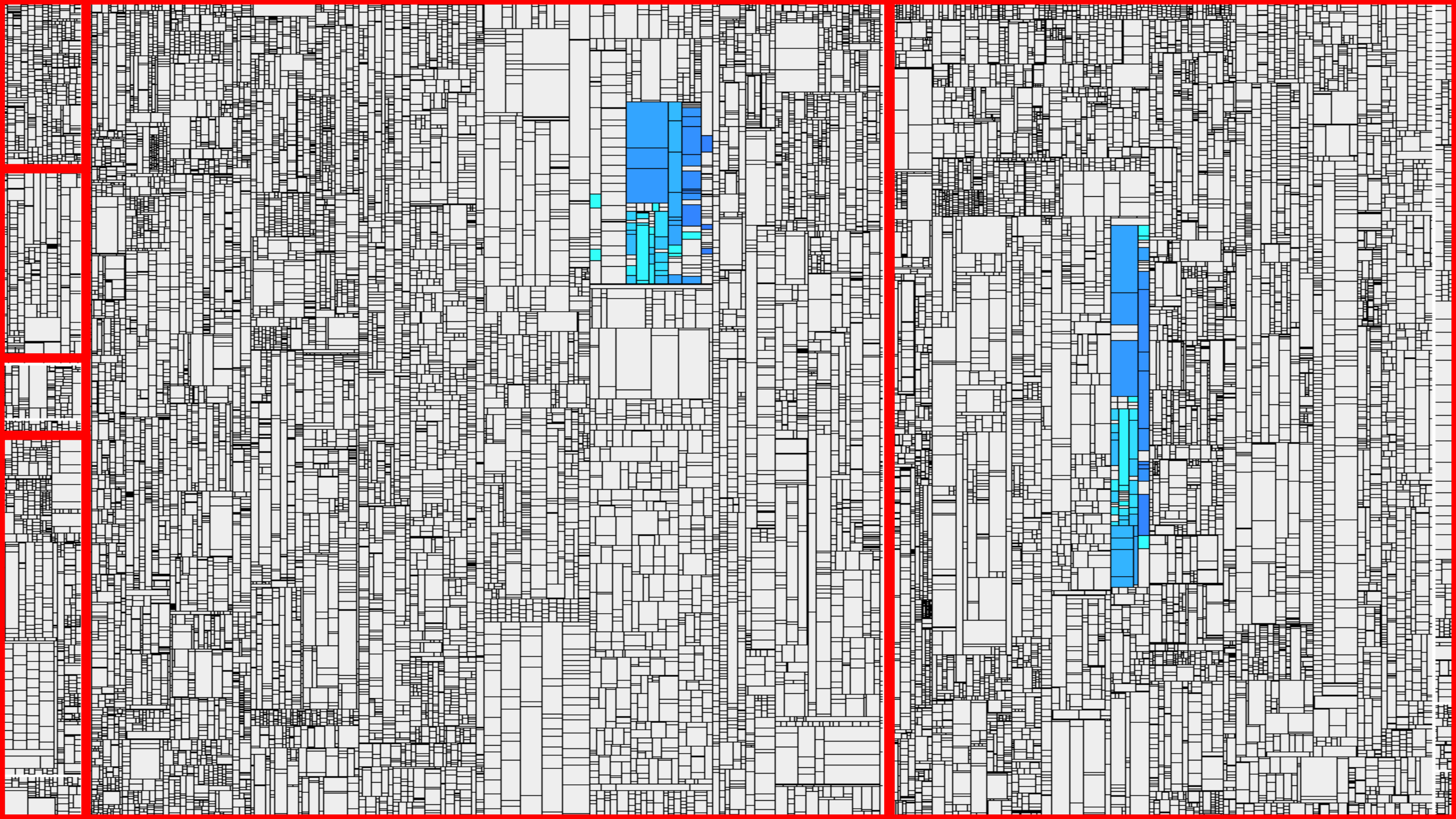
Product 4

Product 5

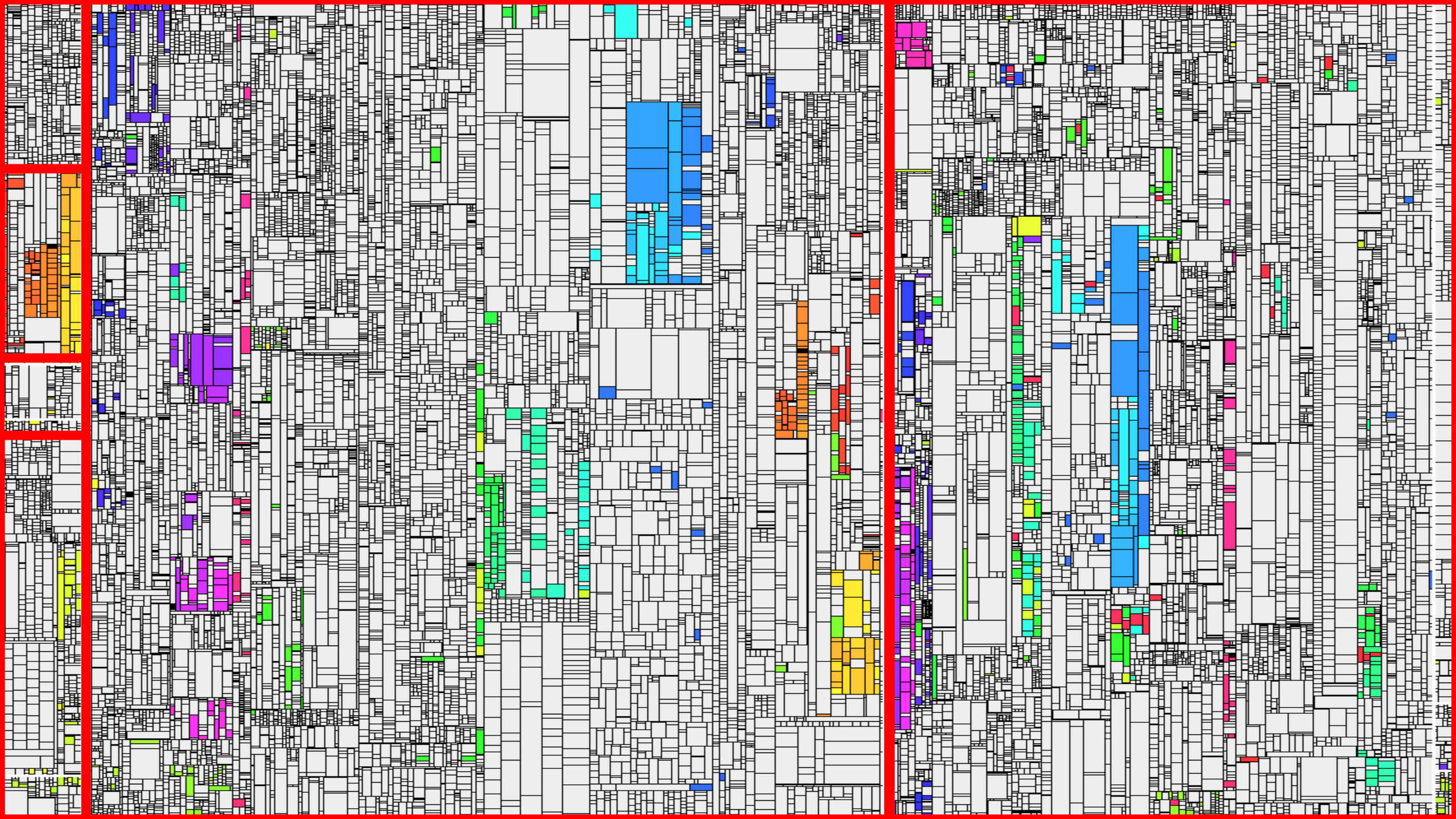


Application Landscape

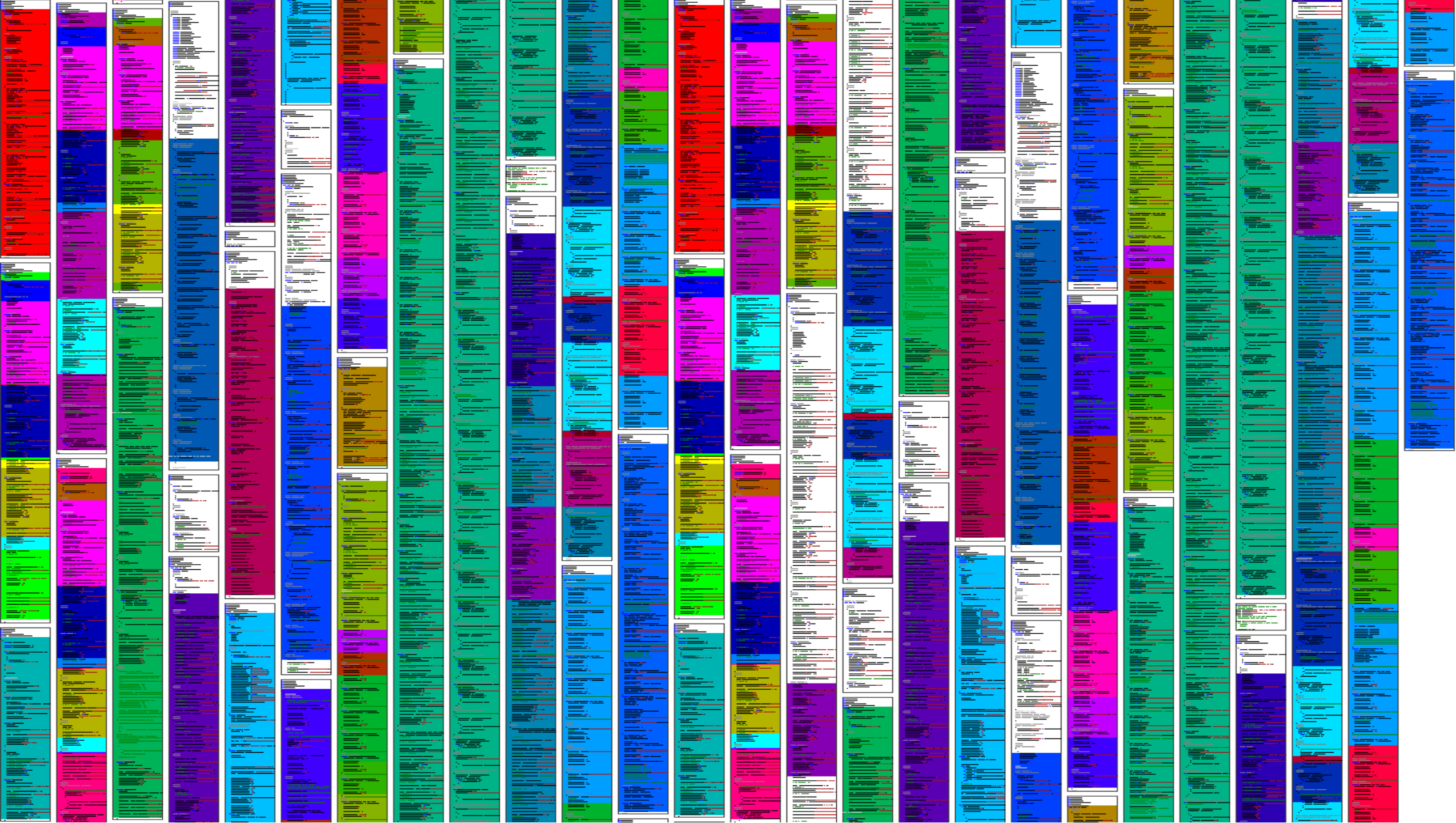




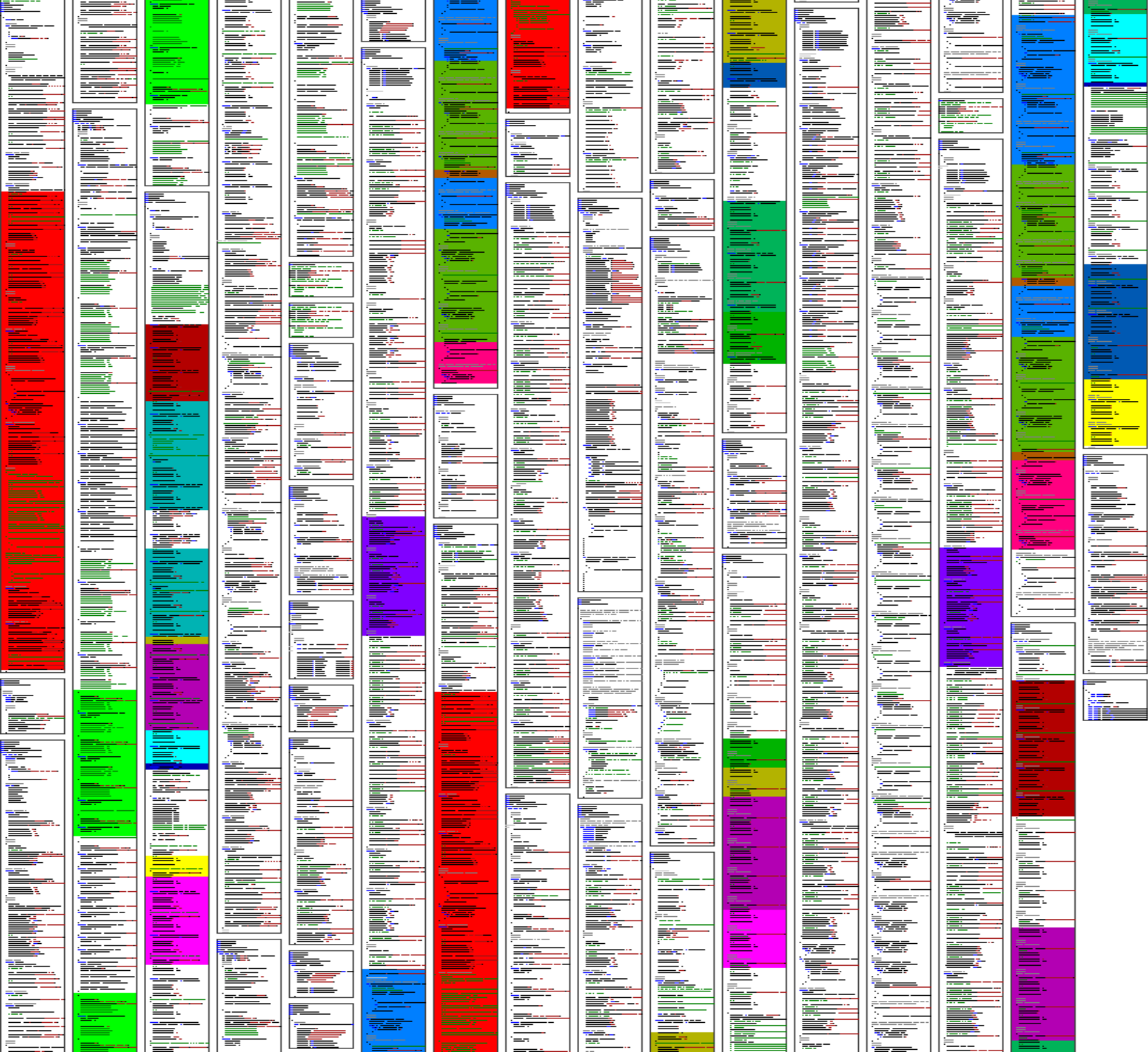




How to resolve this?





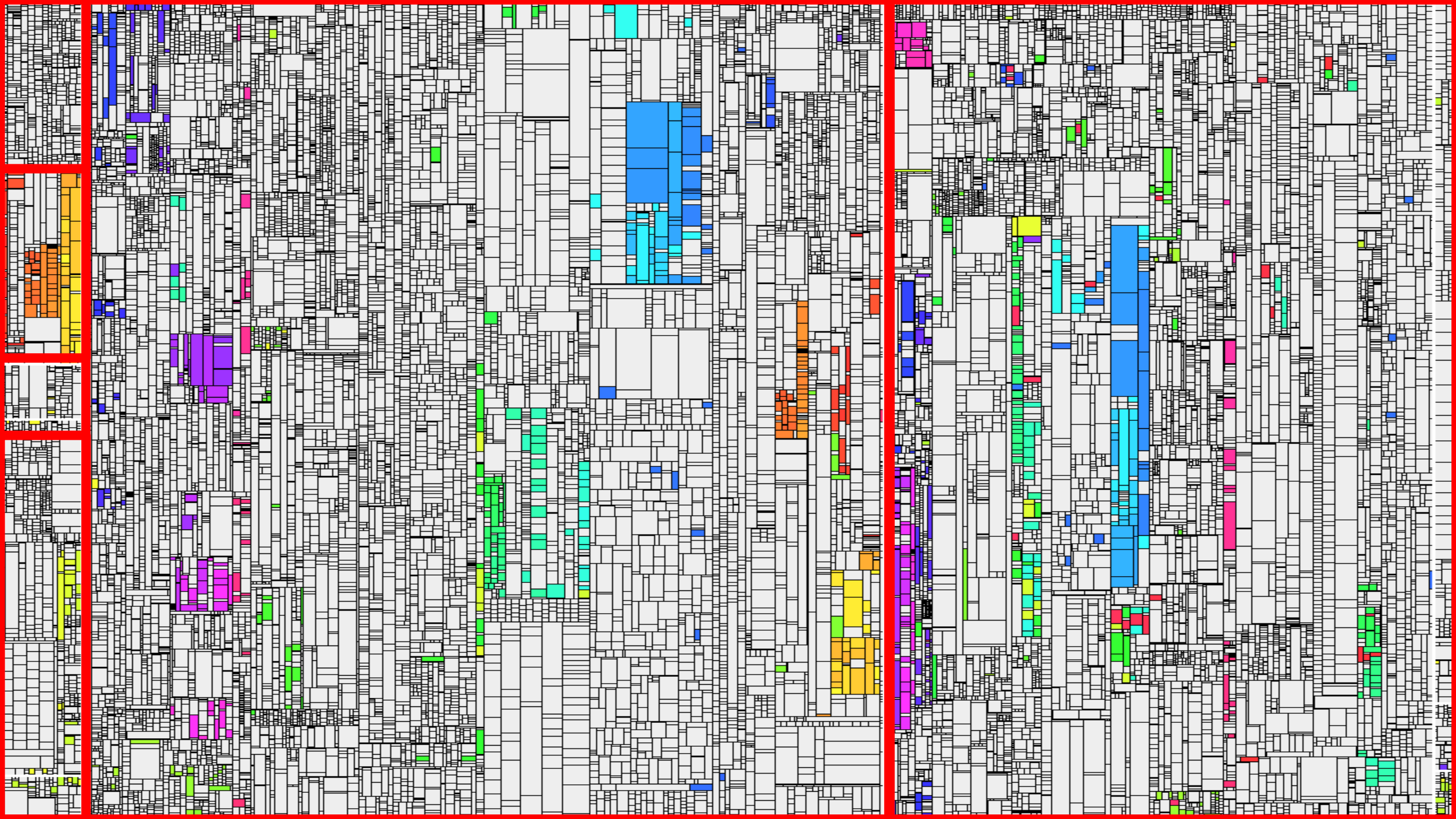


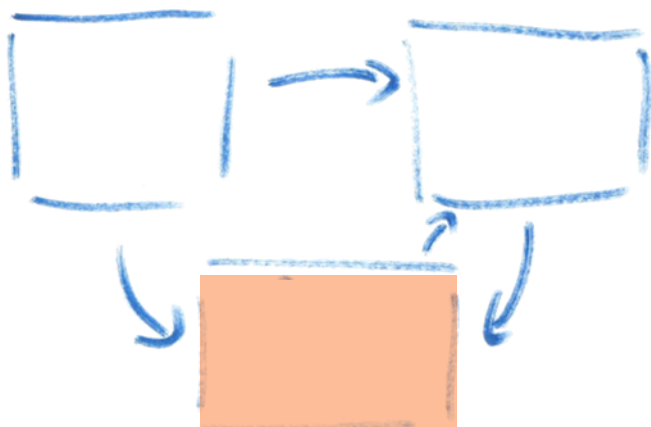
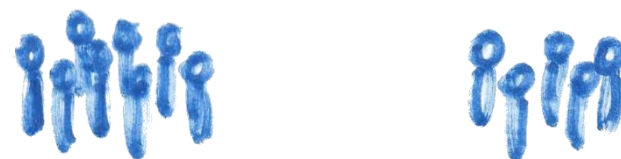
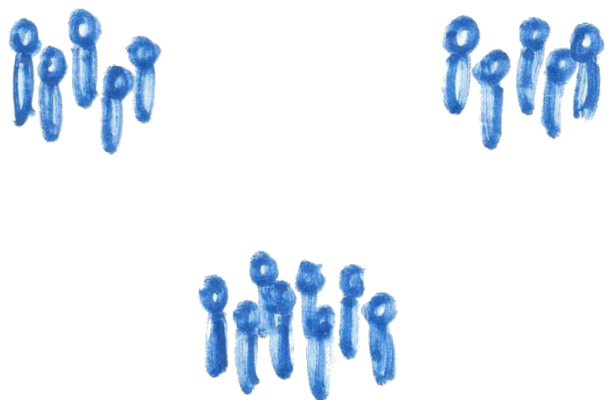
**Buildsystem of one
application landscape**

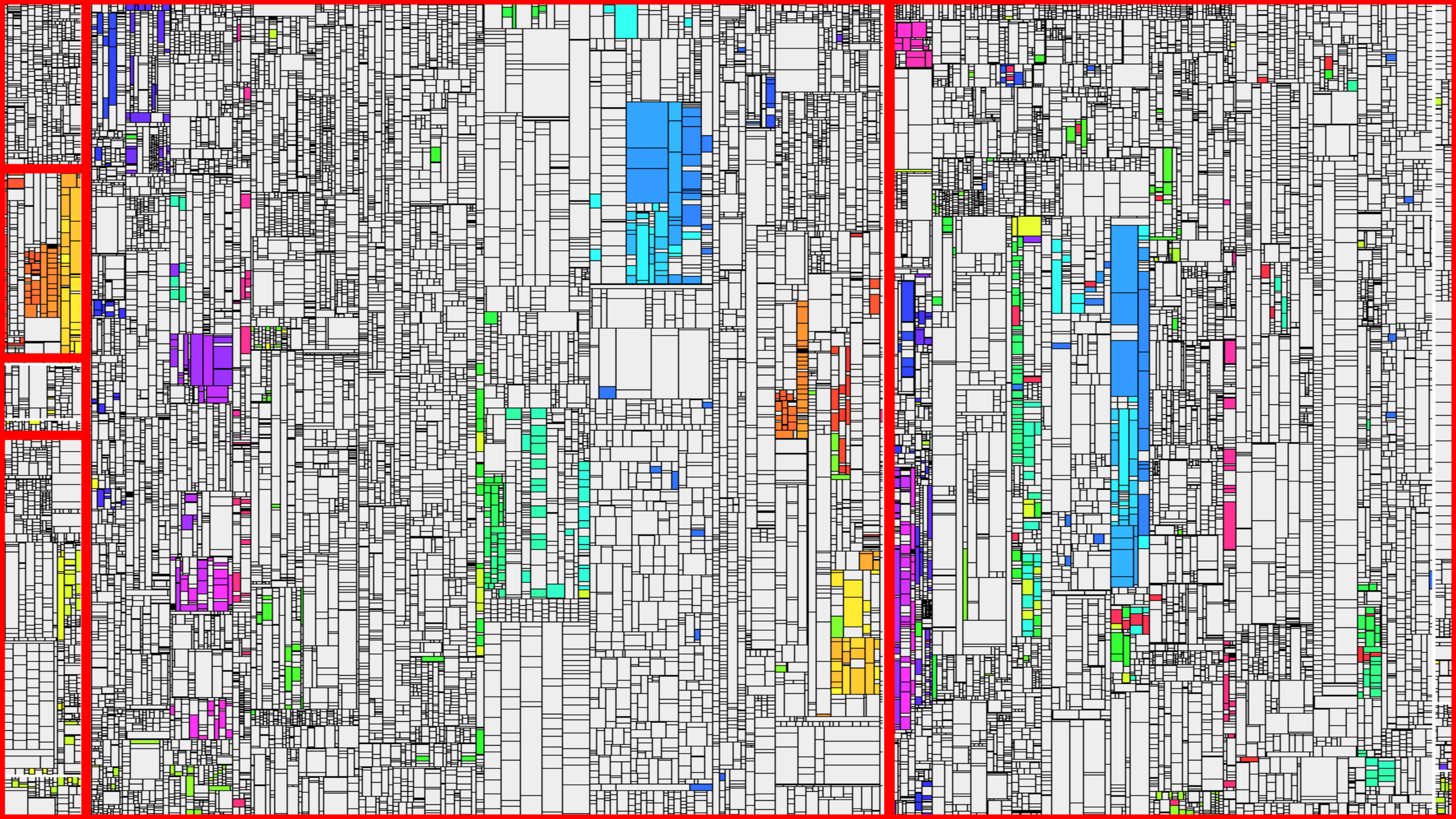
Ad-Hoc Reuse in a Build-System



	Ad hoc Reuse
Build Definitions	81
Lines of Code	457.388
Clone Coverage	97,5%
Blow-Up	929%







```

53 tierQuery.or(\orCriteria -> {
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) {
74     // ...
75   }
76 }

```

```

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) {
75     // ...
76   }
77 }

```



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

Found potential inconsistent clone change in PermissionHandler.java

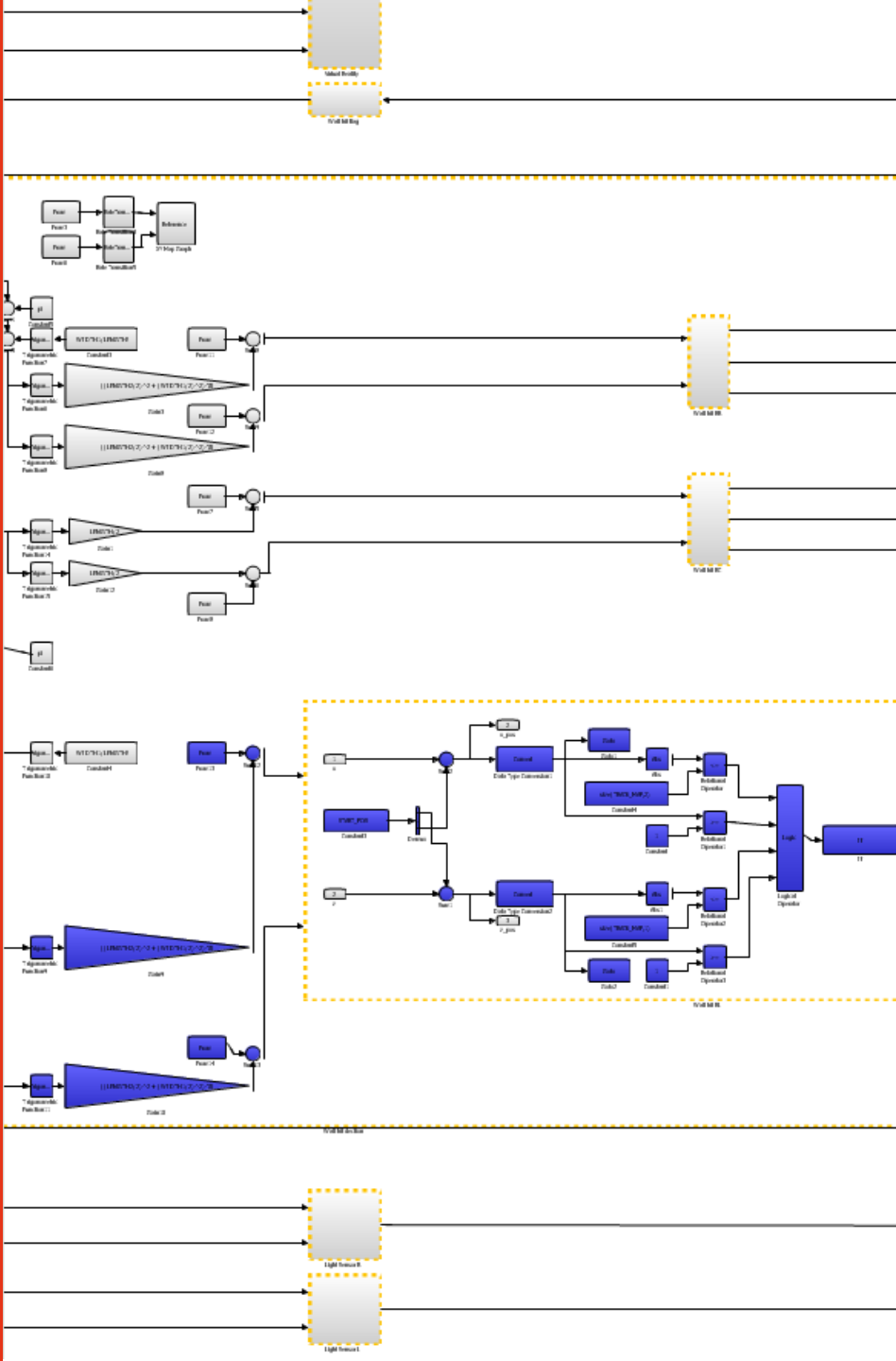
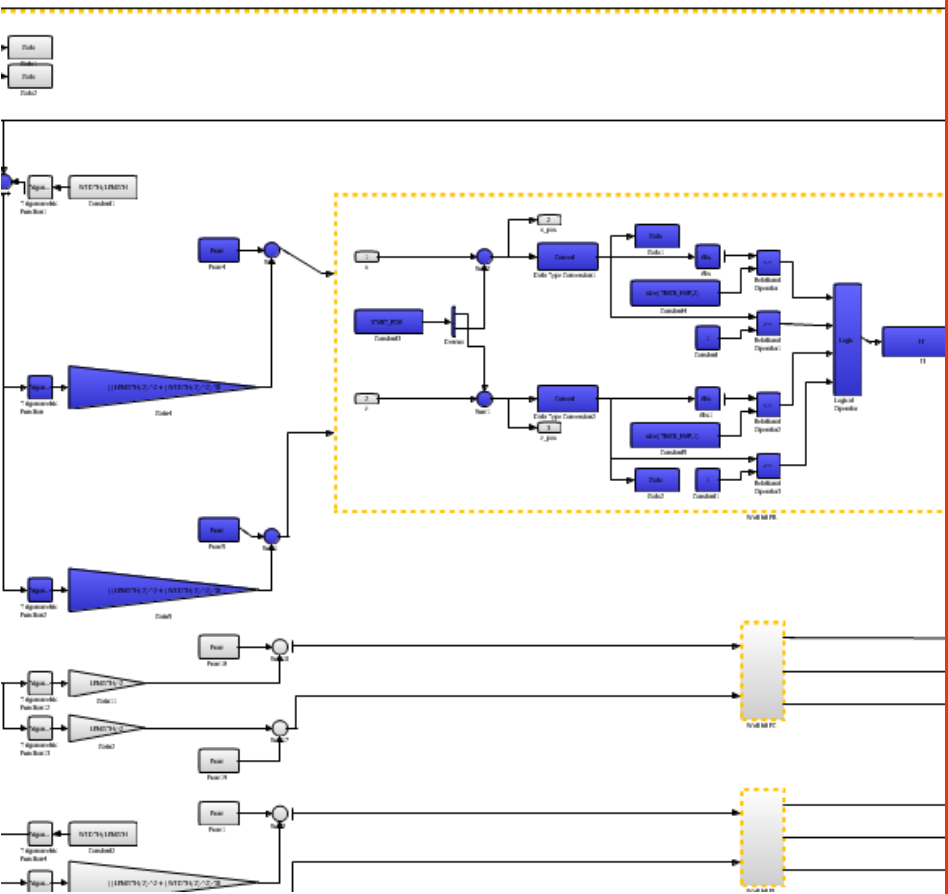
Context

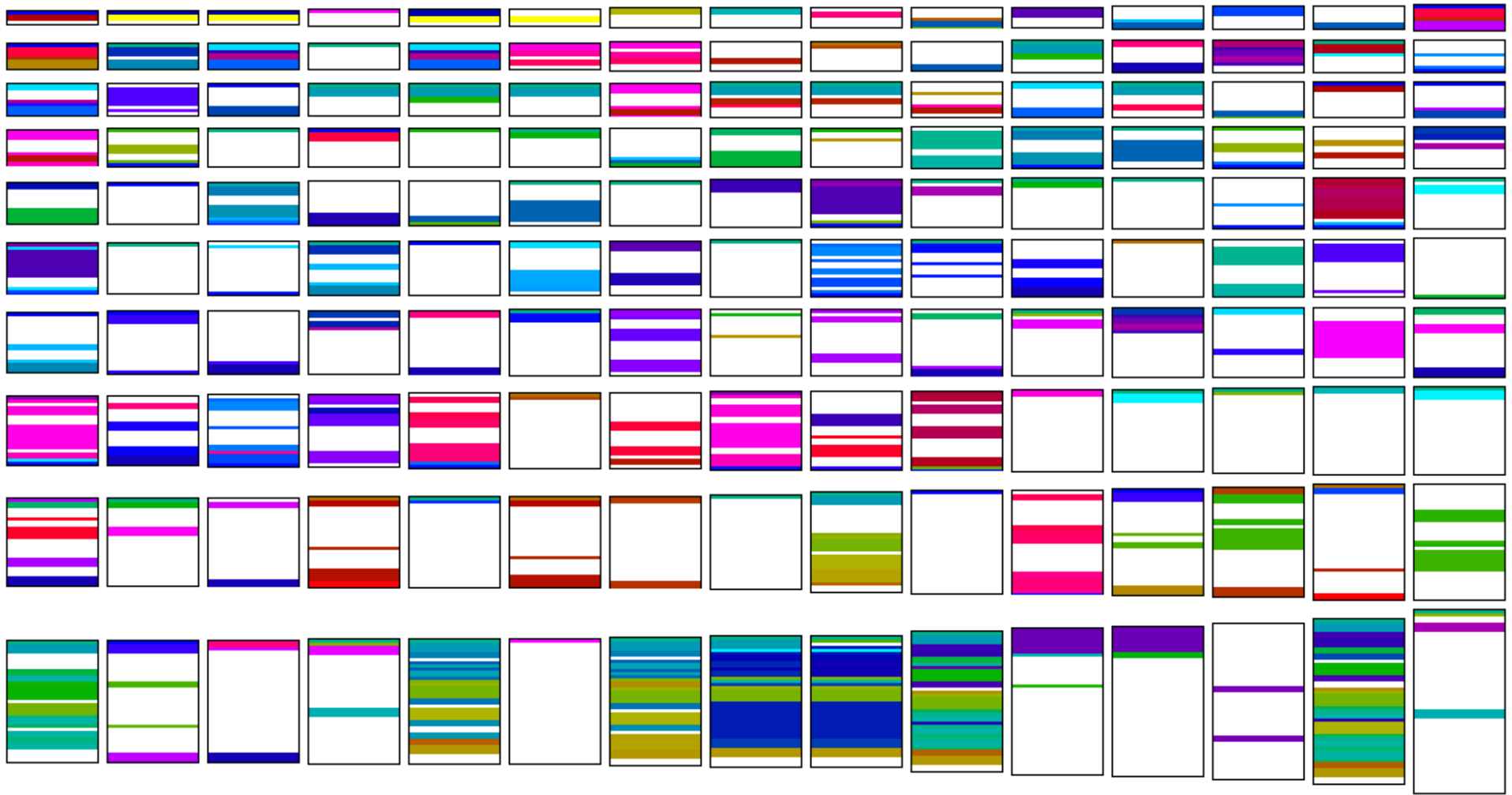
Broken clone

Old clone finding

Code change

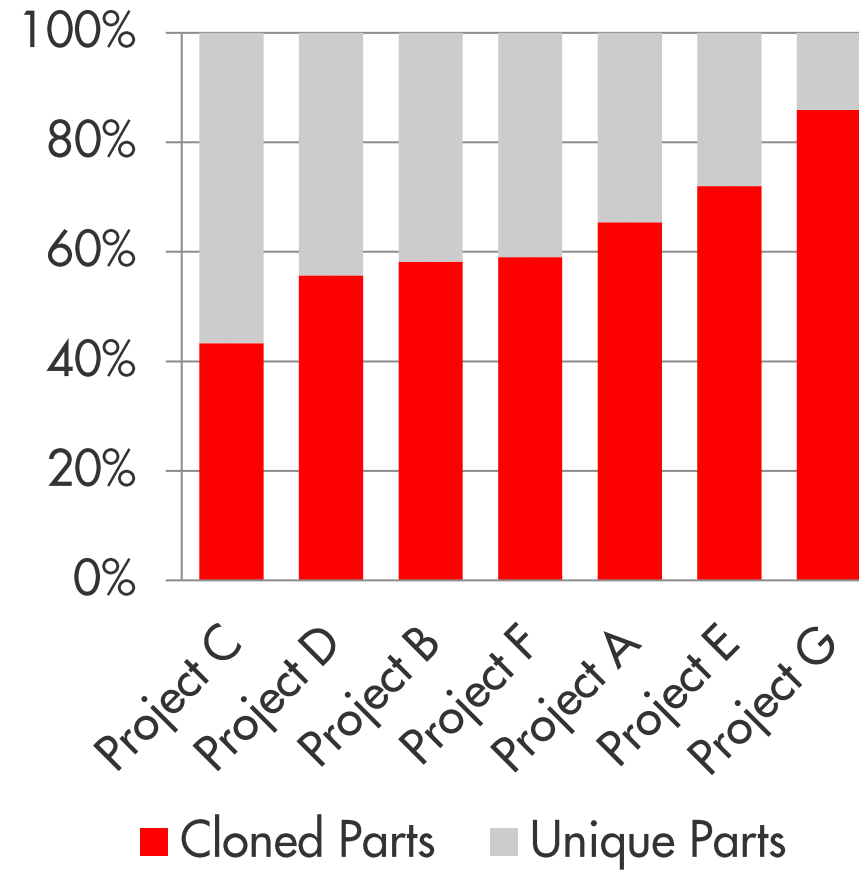
**What about software
artifacts other than code?**





176 use cases in total, 150 contain cloning

	System under Test	Test Suite
		# Tests
System A	330 kLoC	266
System B	580 kLoC	1,059
System C	150 kLoC	72
System D	430 kLoC	180
System E	760 kLoC	1,804
System F	1,400 kLoC	135
System G	160 kLoC	605



**Was all of this
duplication really
copied?**



Splitting with java.lang.String.split()

```
String[] addresse2 = addresse.split(Pattern.  
    quote(String.valueOf(separator)));
```

Splitting with java.util.StringTokenizer

```
ArrayList<String> validEmails = new ArrayList<  
    String>();  
StringTokenizer st = new StringTokenizer(  
    addresse, Character.toString(separator));  
while (st.hasMoreTokens()) {  
    String tmp = st.nextToken();  
    validEmails.add(tmp);  
}
```

Splitting with custom algorithm 1

```
List<String> result = new ArrayList<String>();  
int z = 0;  
for (int i=0; i<addresse.length(); i++) {  
    if (i==addresse.length()-1) {  
        result.add(addresse.substring(z, i+1));  
    }  
    if (addresse.charAt(i)==separator) {  
        result.add(addresse.substring(z, i));  
        z=i+1;  
    }  
}
```

Splitting with custom algorithm 2

```
List<String> curAddrs = new ArrayList<String>();  
String buffer = "";  
for(int i=0; i<addresse.length(); i++) {  
    if(addresse.charAt(i) != separator) {  
        buffer += addresse.charAt(i);  
    } else {  
        curAddrs.add(buffer);  
        buffer = "";  
    }  
}  
curAddrs.add(buffer);
```

Splitting with custom algorithm 3

```
List<String> emailListe= new ArrayList<String>();  
int trenneralt = 0;  
while (addresse.indexOf(separator, trenneralt) !=  
    -1) {  
    int trennerneu = addresse.indexOf(separator,  
        trenneralt);  
    emailListe.add(addresse.substring(trenneralt,  
        trennerneu));  
    trenneralt = trennerneu + 1;  
}
```

Summary

Ad-Hoc Reuse (Copy&Paste) happens often in practice. In all programming languages and also in other software artifacts (other than code).

In some cases, management of existing copies is more pragmatic than their removal. Most teams are unaware, however, that clone management tools exist.

For new systems, each large copy should raise the question of a better alternative.

Clone detection is a good tool to quickly find duplication. Not only in code.



Feb. 01

Learn and Get Paid - Work as a Remote Software Development Student at CQSE

Learn professional software development at CQSE. Work on our tool Teamscale with our team remotely and get helpful feedback on every change!

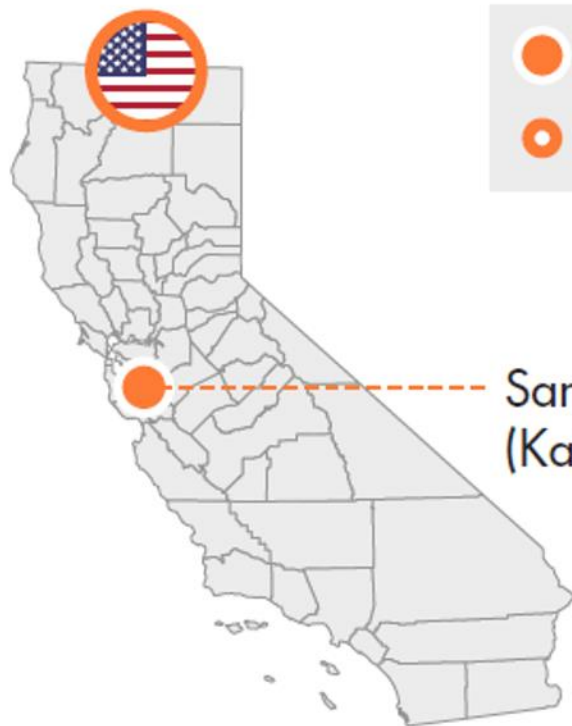
<https://cqse.eu/lagp-2023-sb>



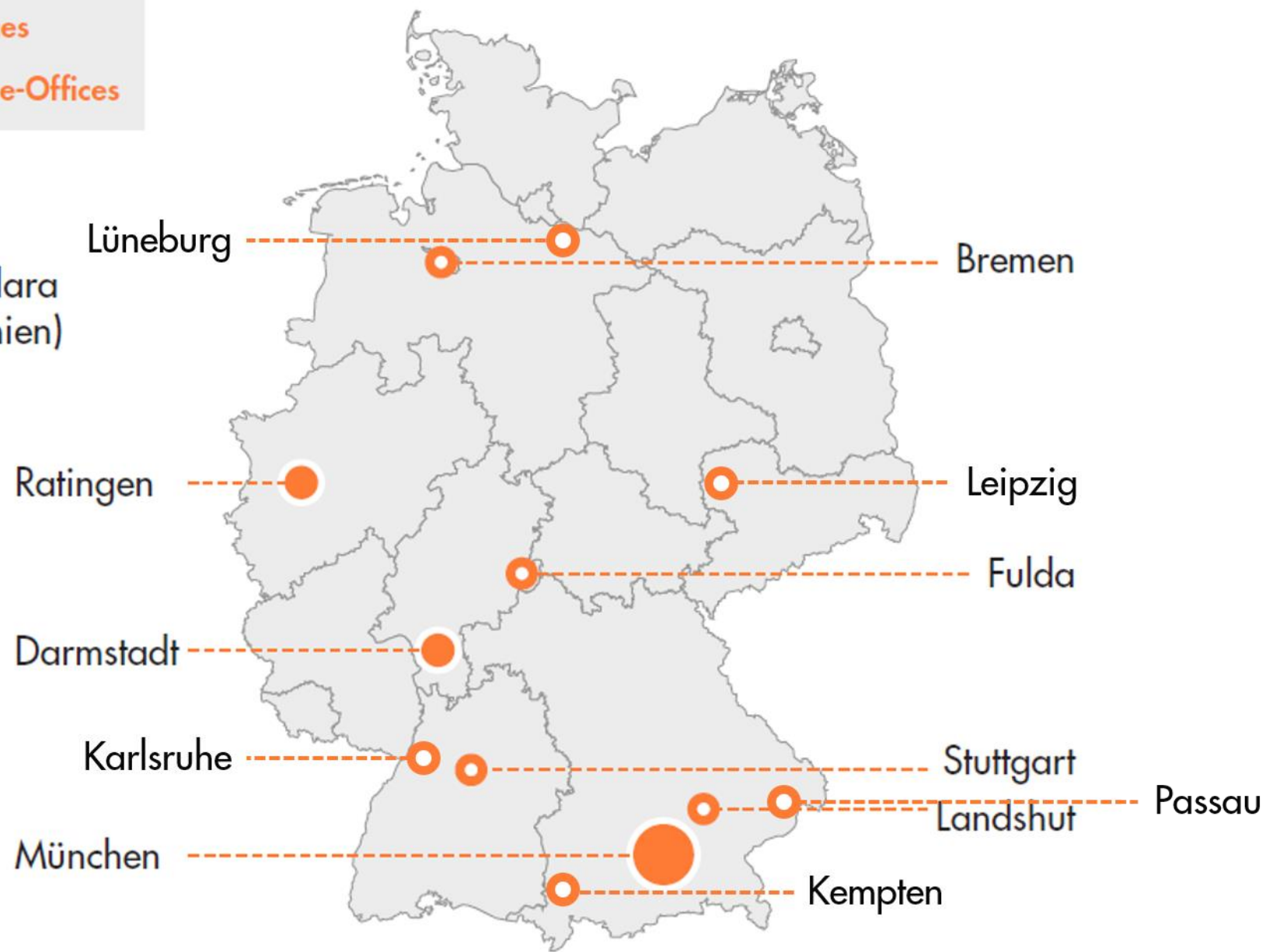
Datum und Uhrzeit

Mi. 1. Februar 2023,
15:30 Uhr – 17:00 Uhr MEZ





- Offices
- Home-Offices



Teamscale ist free for Research & Teaching

Reach out 😊: juergens@cqse.eu

Kontakt – Looking forward to Discussions 😊



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

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

