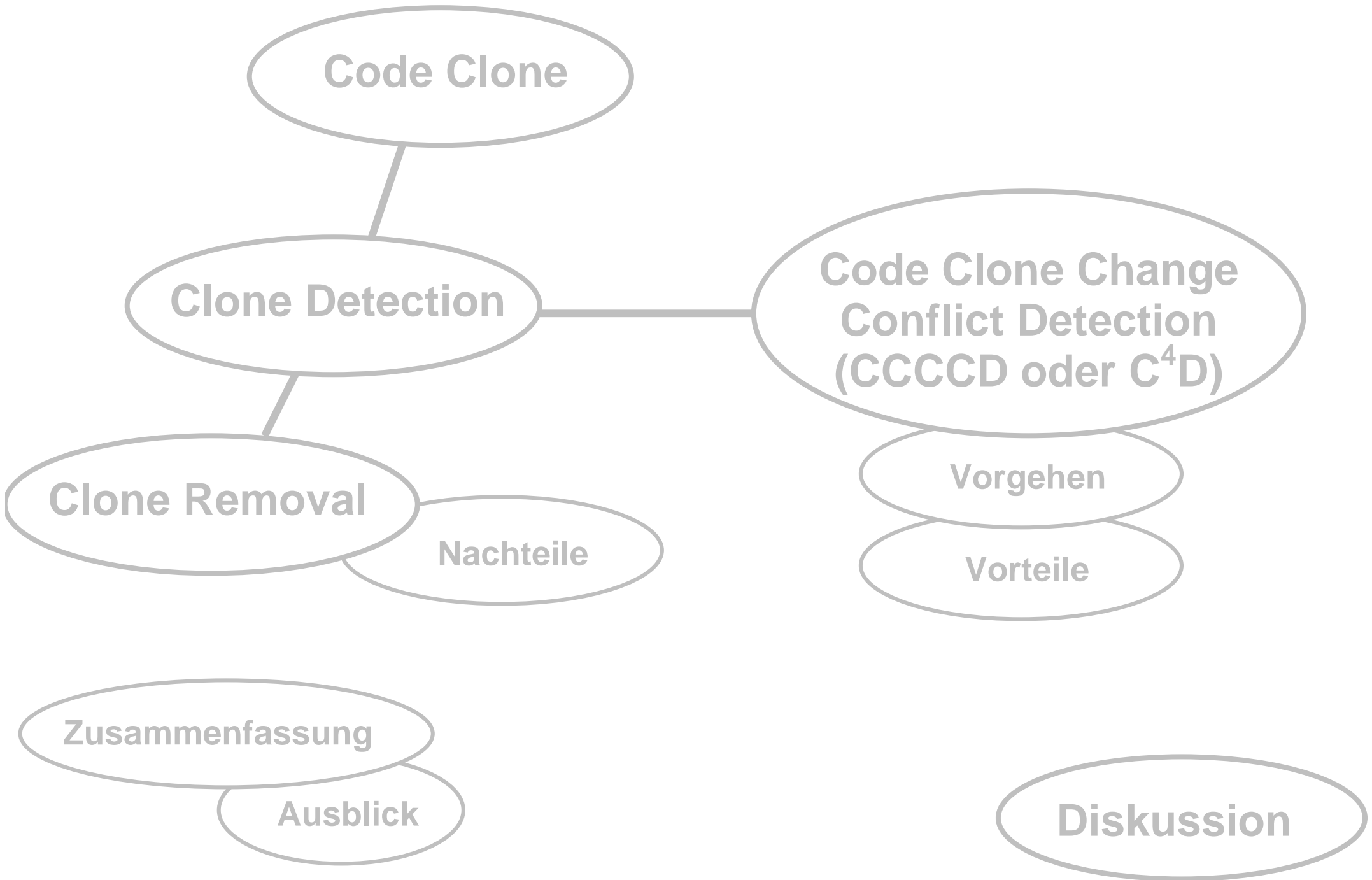


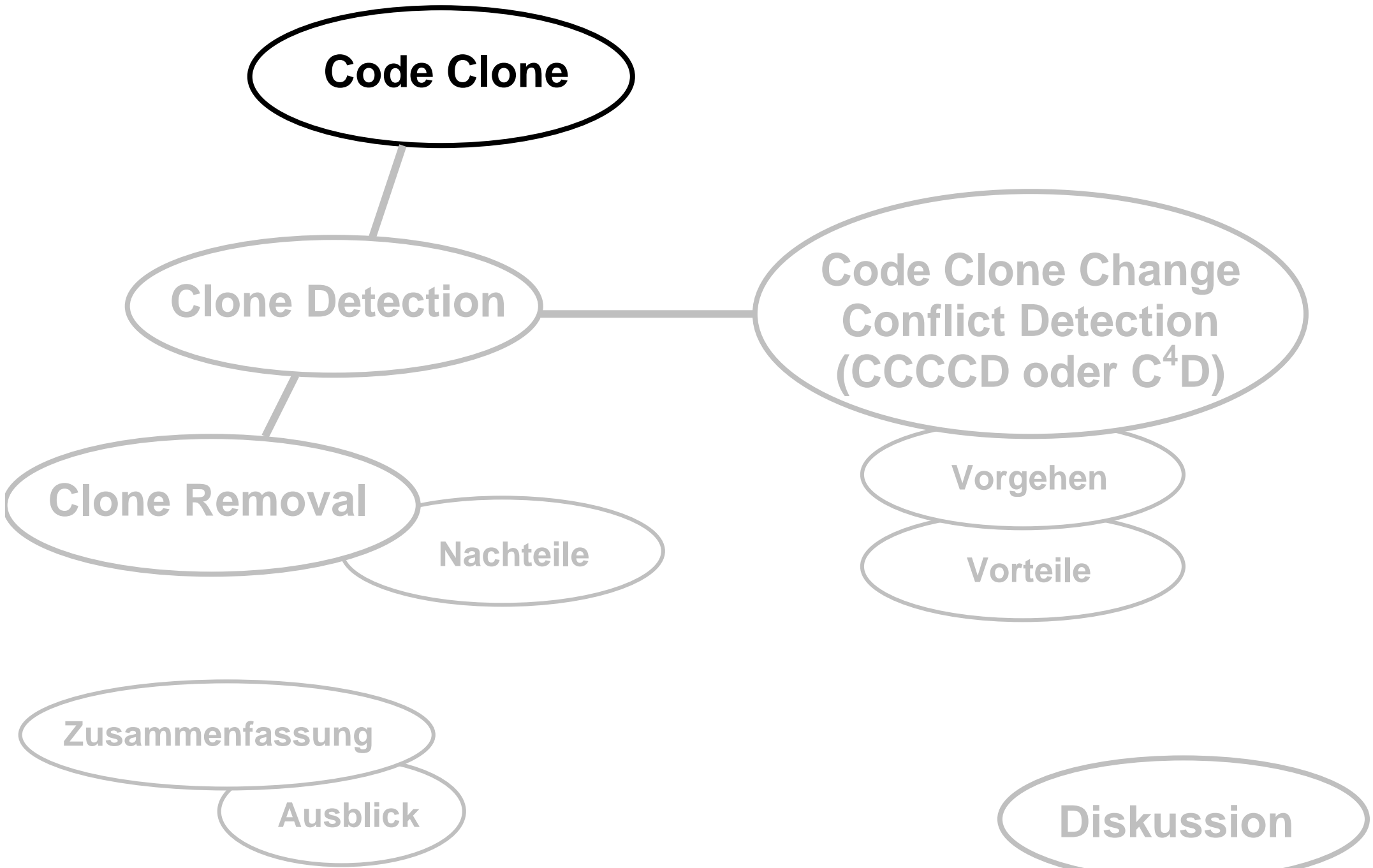
**C<sup>4</sup>D**

**oder**

**Wie ich lernte, mit Code Clones zu leben**

Udo Borkowski – IT Consulting – Aachen





```

public static String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }

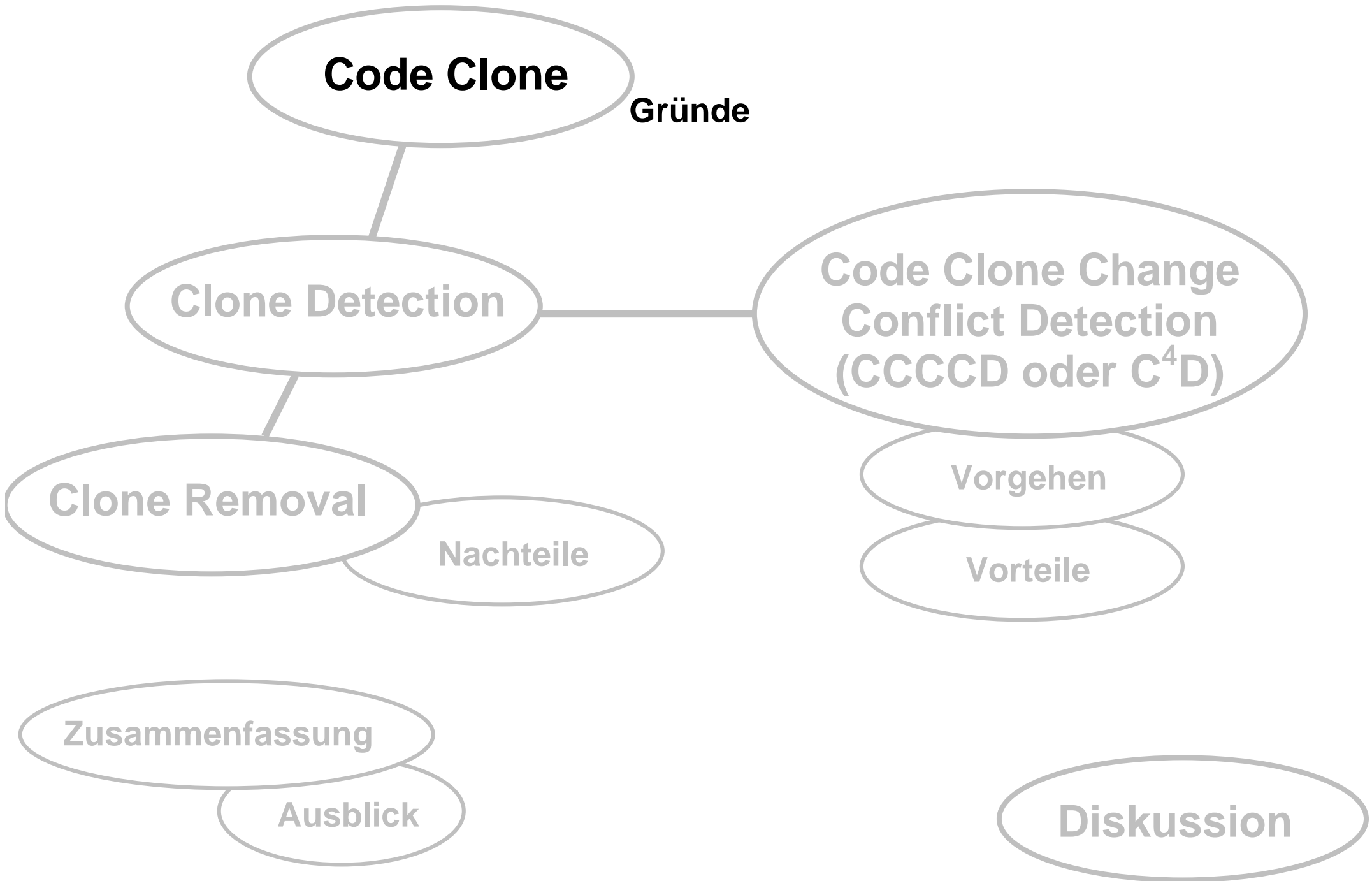
        File file = new File("../..../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        Log.error\("ReportService",
            "Error finding engine.xml: " + ex.getMessage());
        return(null);
    }
}

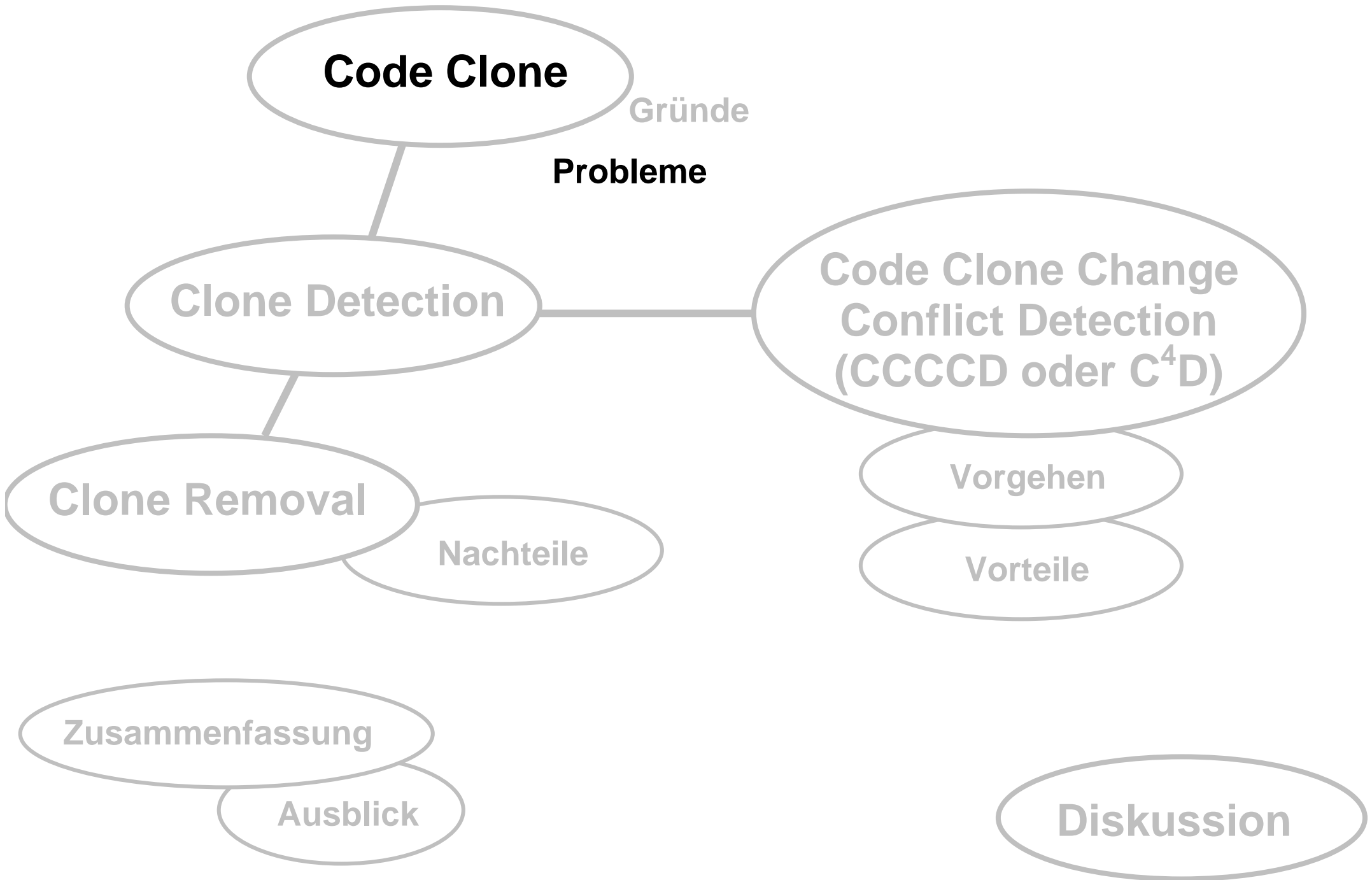
```

```

public String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }
        ServletContext sc = getServletContext\(\);
        String path =
            sc.getInitParameter\("company.conf"\);
        if (path != null) {
            return\(path\);
        }
        File file = new File("../..../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        System.err.println\(
            "Error finding engine.xml: " +
            ex.getMessage());
        return(null);
    }
}

```





```

public static String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }

        File file = new File("../..../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        Log.error("ReportService",
            "Error finding engine.xml: " + ex.getMessage());
        return(null);
    }
}

```

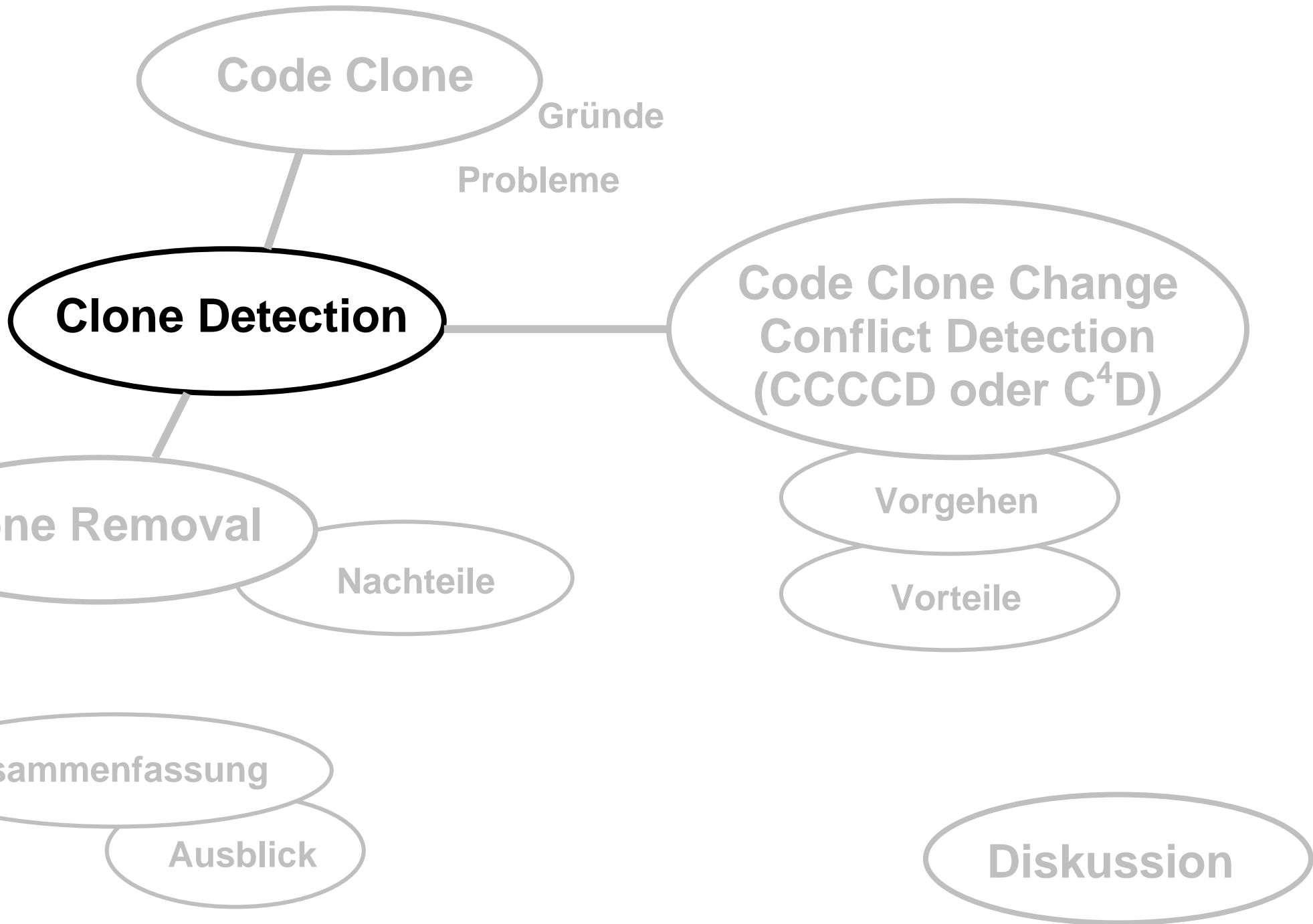
```

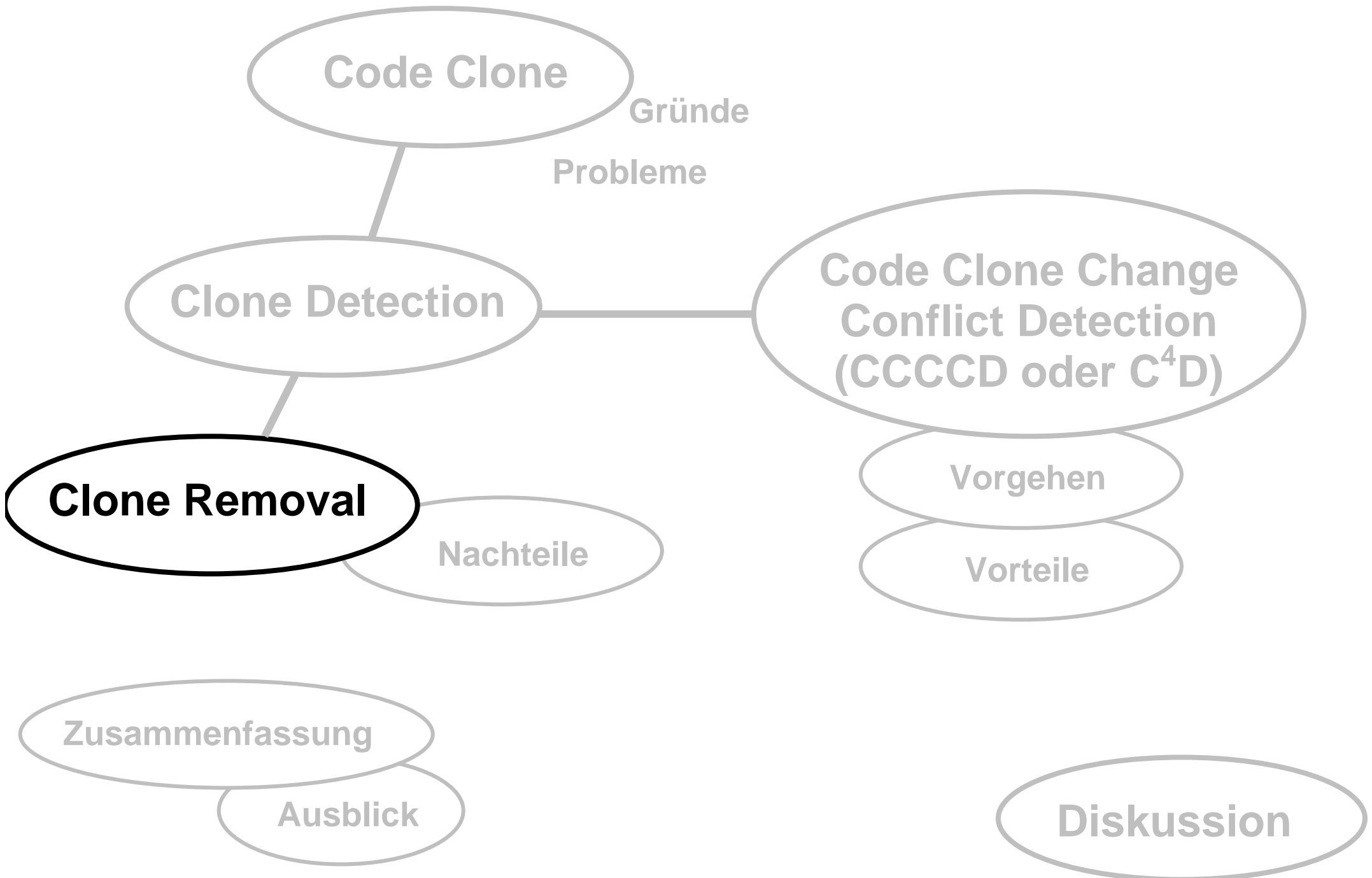
public String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }
        ServletContext sc = getServletContext();
        String path =
            sc.getInitParameter("company.conf");
        if (path != null) {
            return(path);
        }
        File file = new File("../..../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        System.err.println(
            "Error finding engine.xml: " +
            ex.getMessage());
        return(null);
    }
}

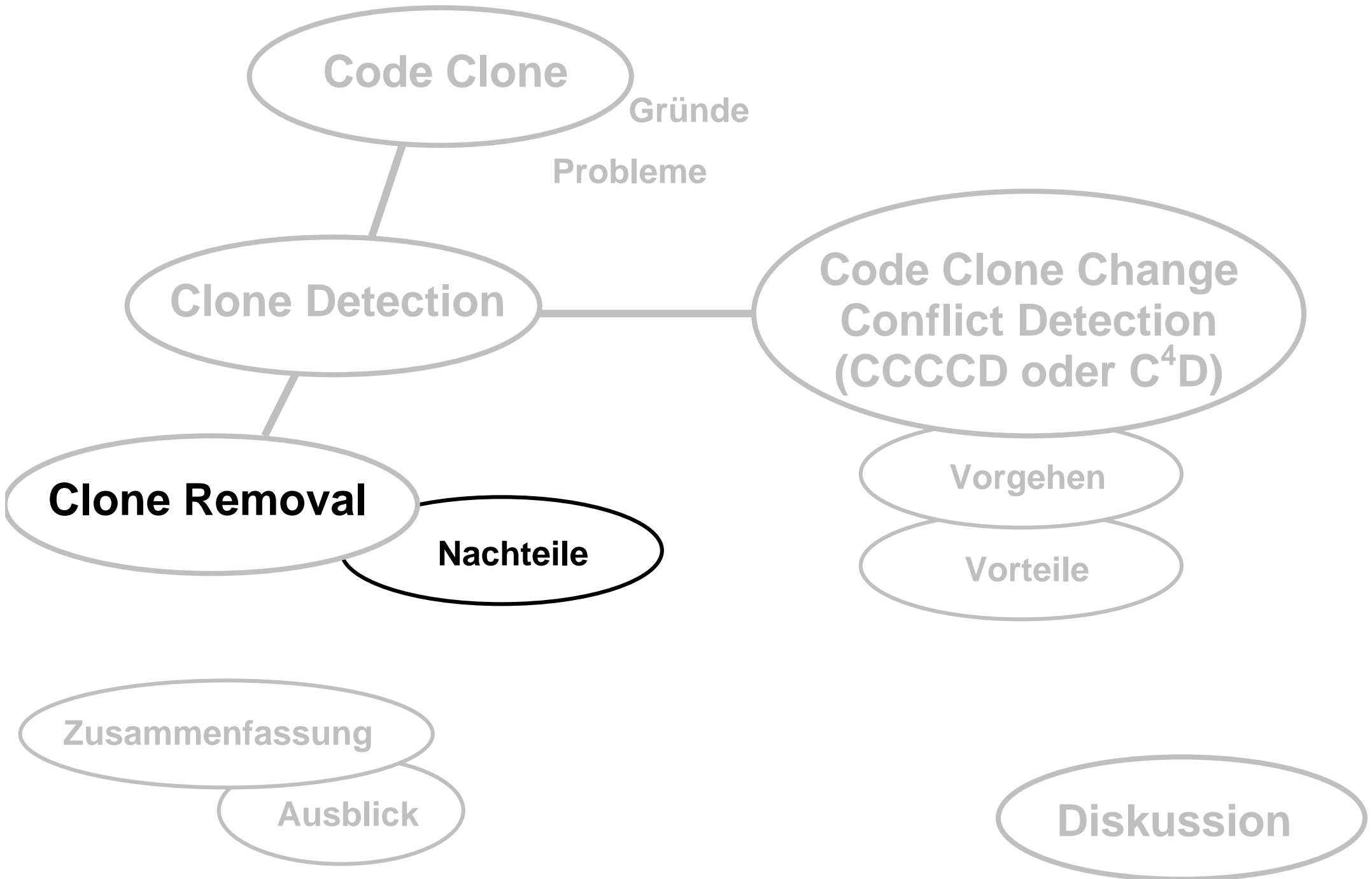
```

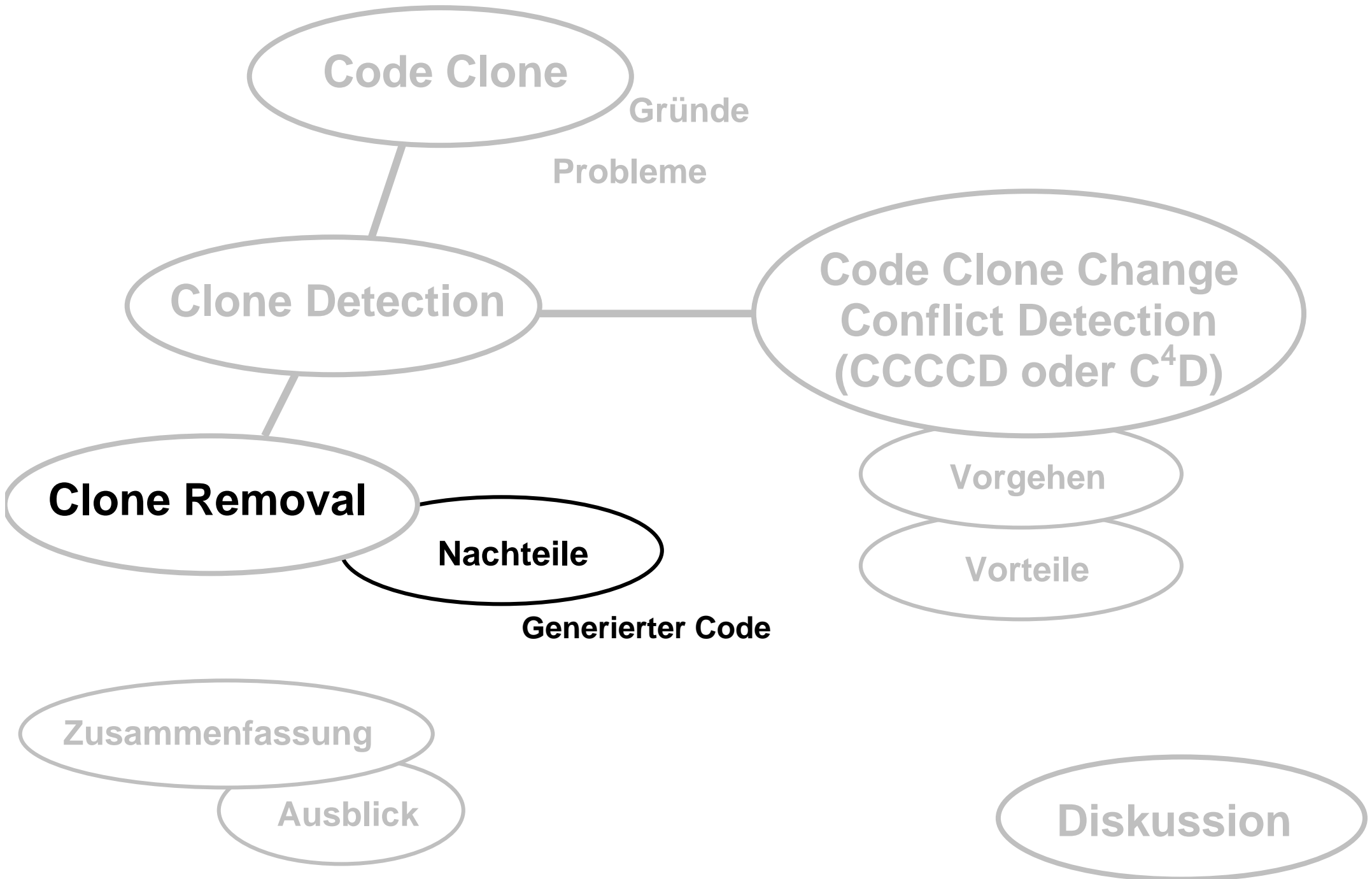


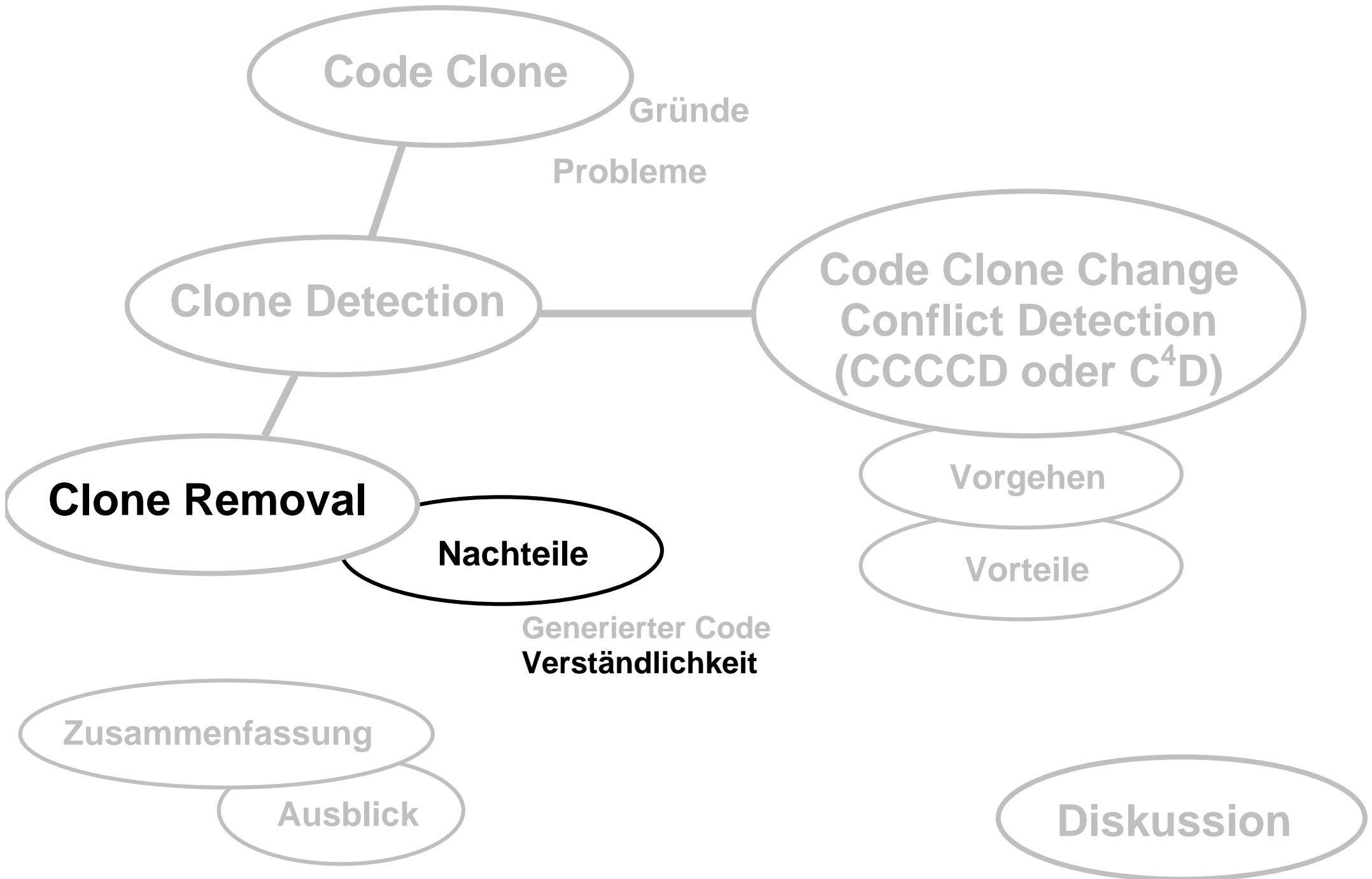


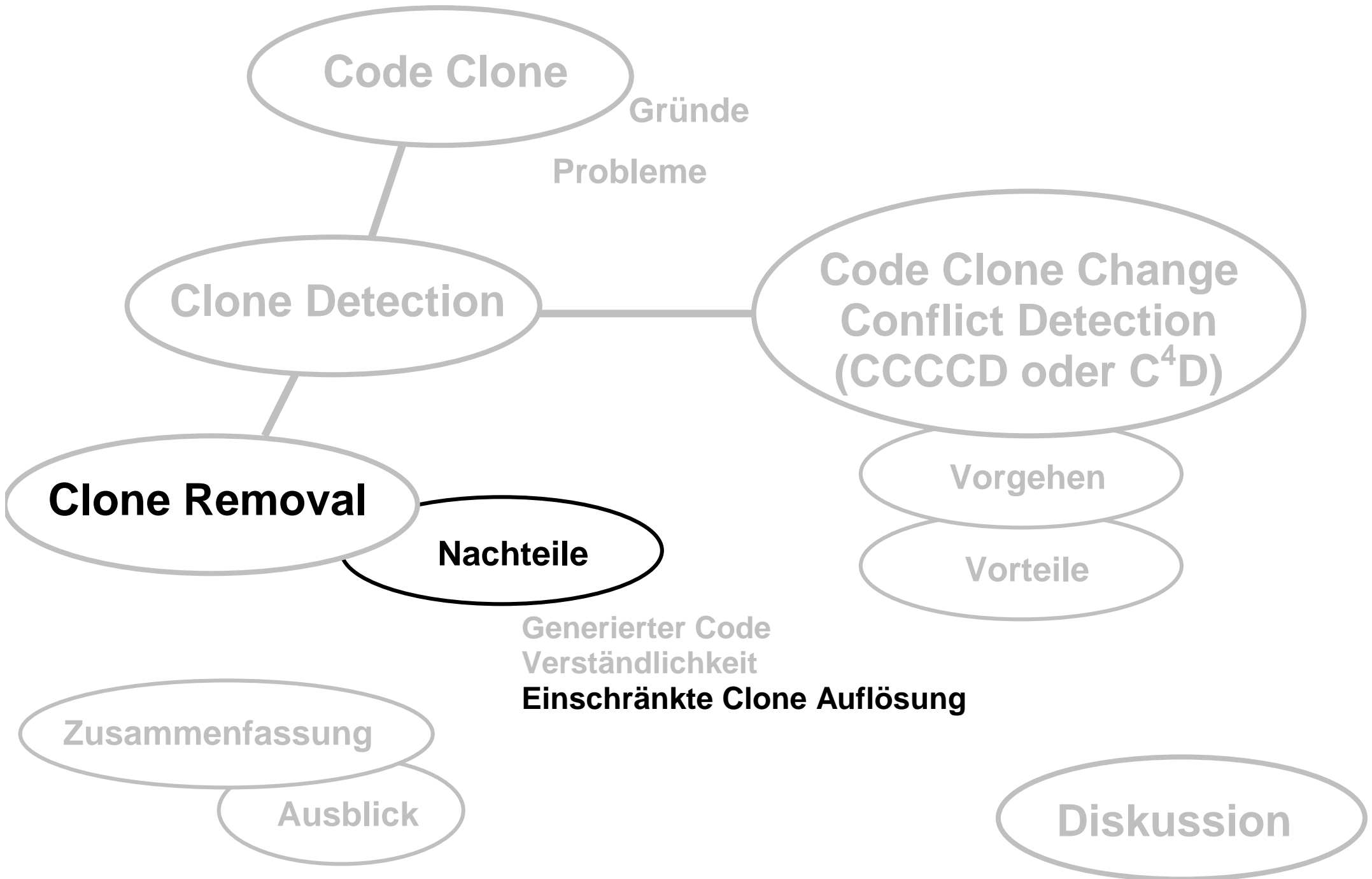


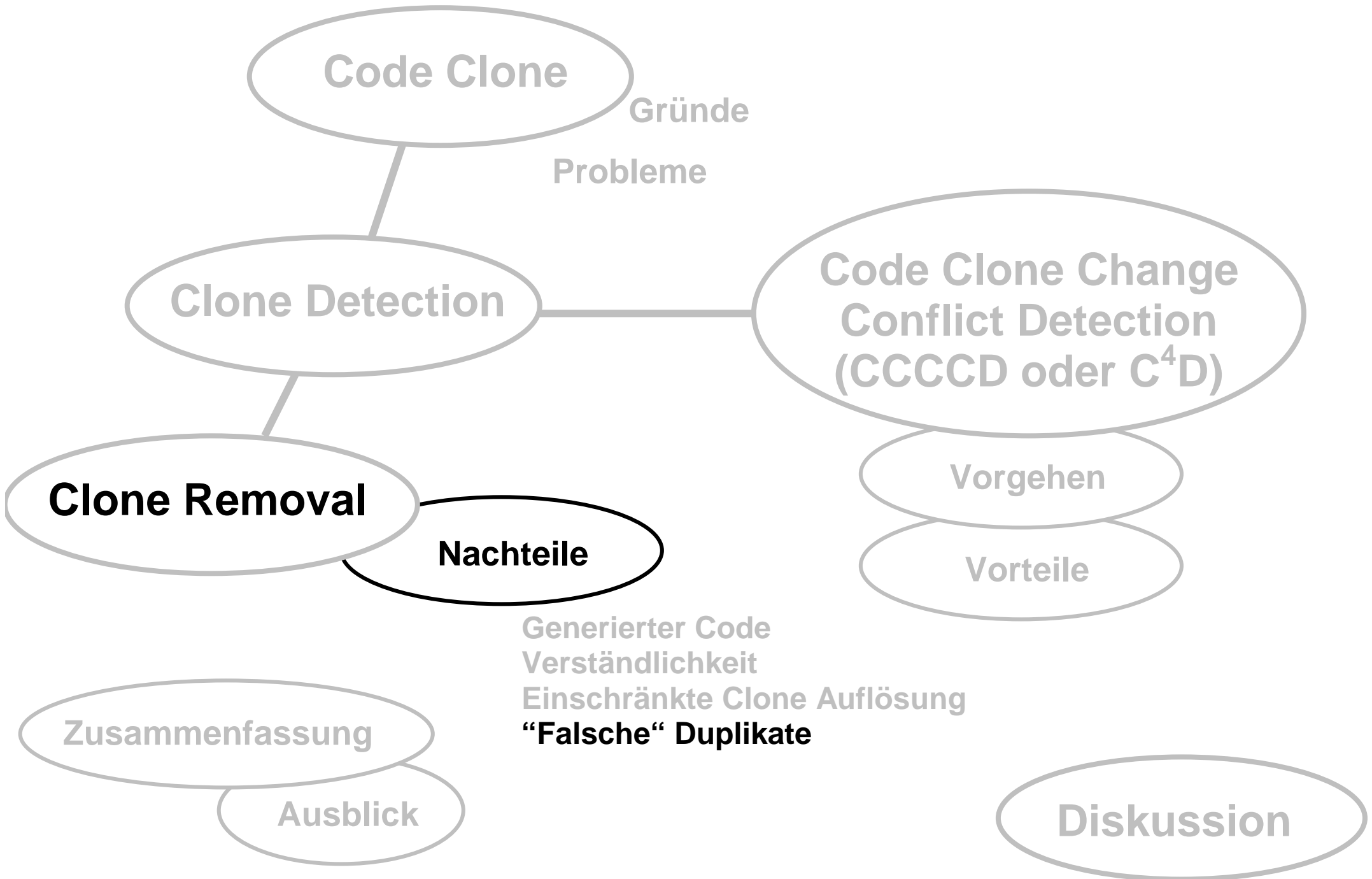


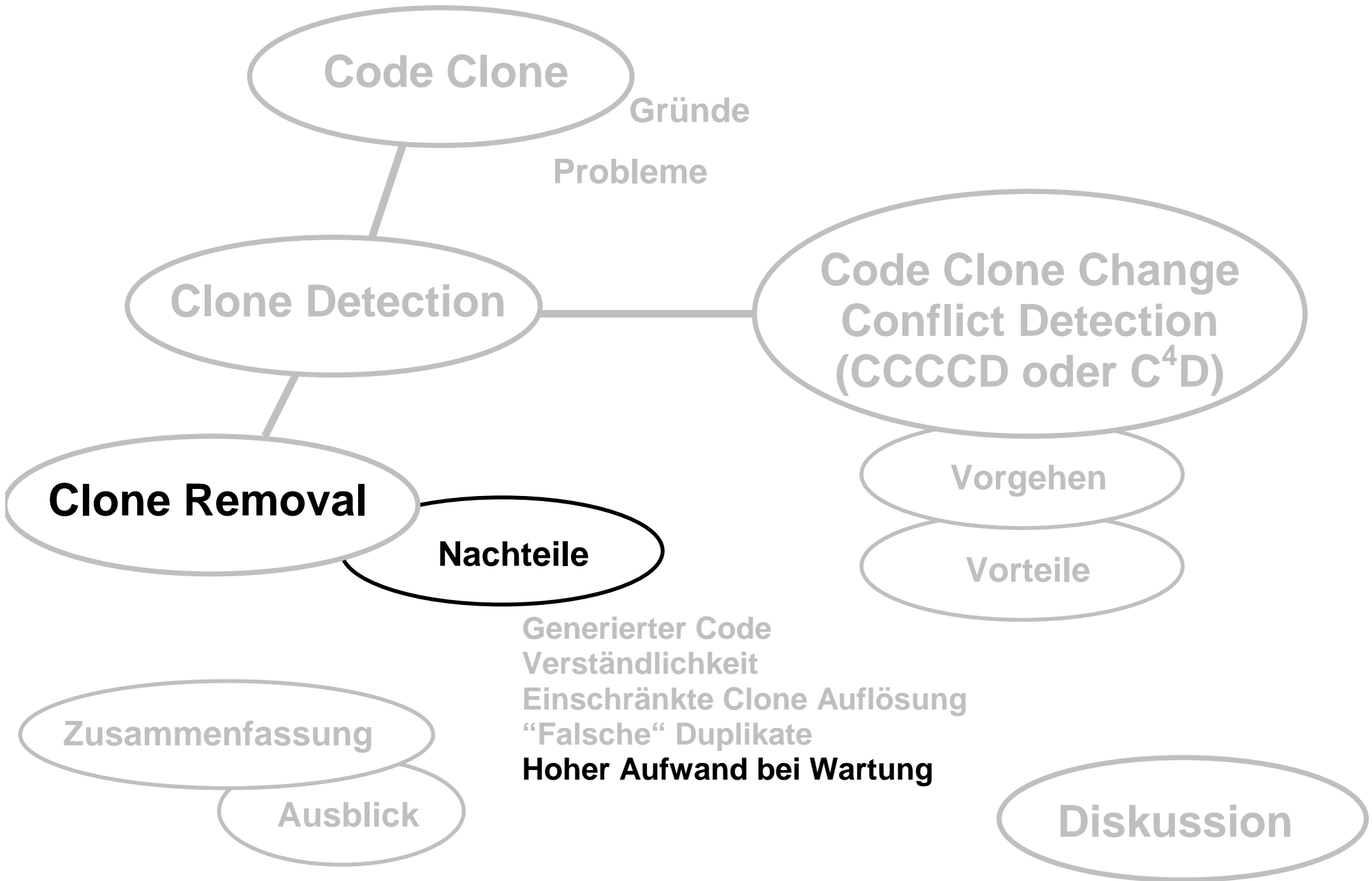




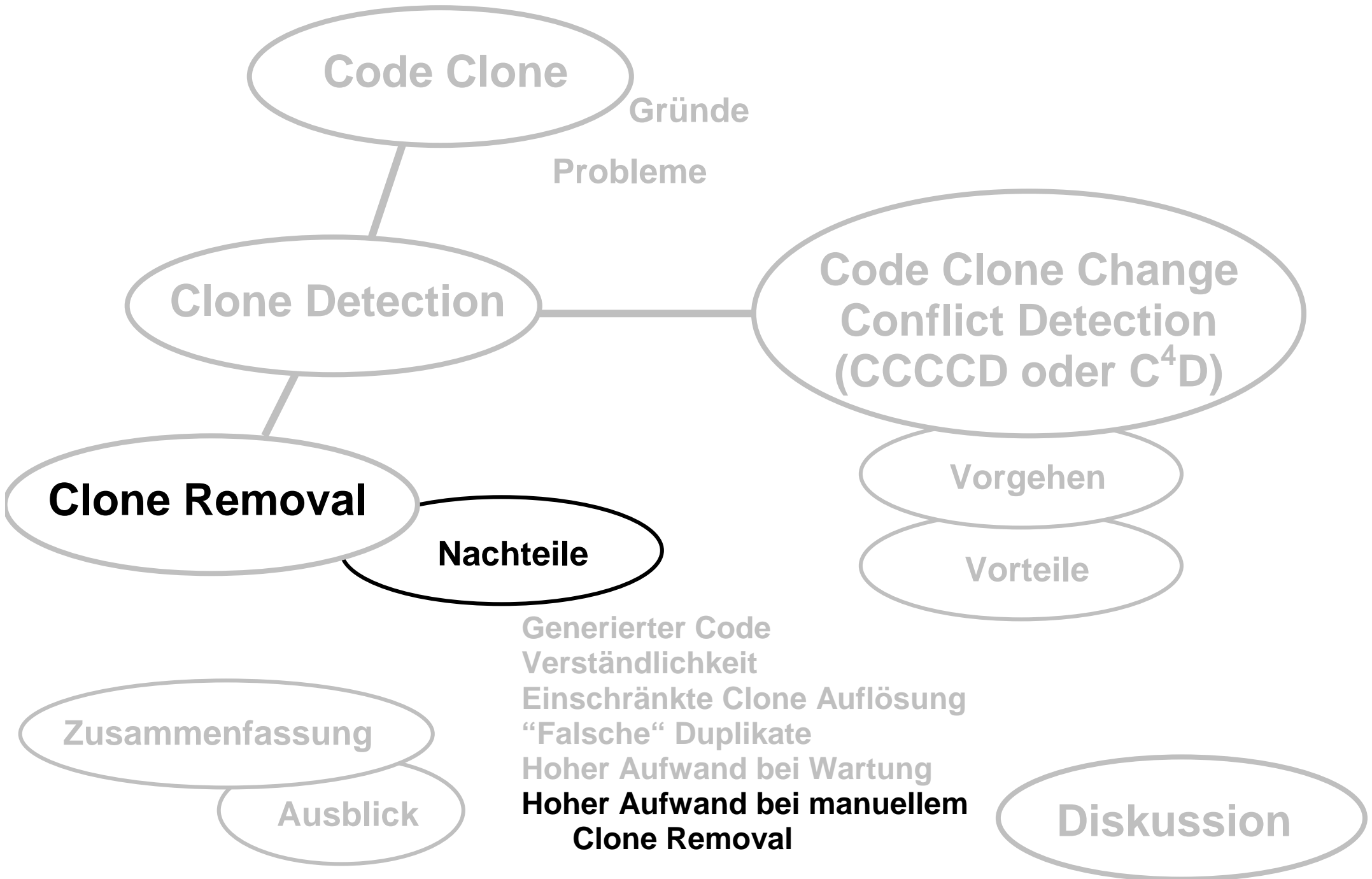


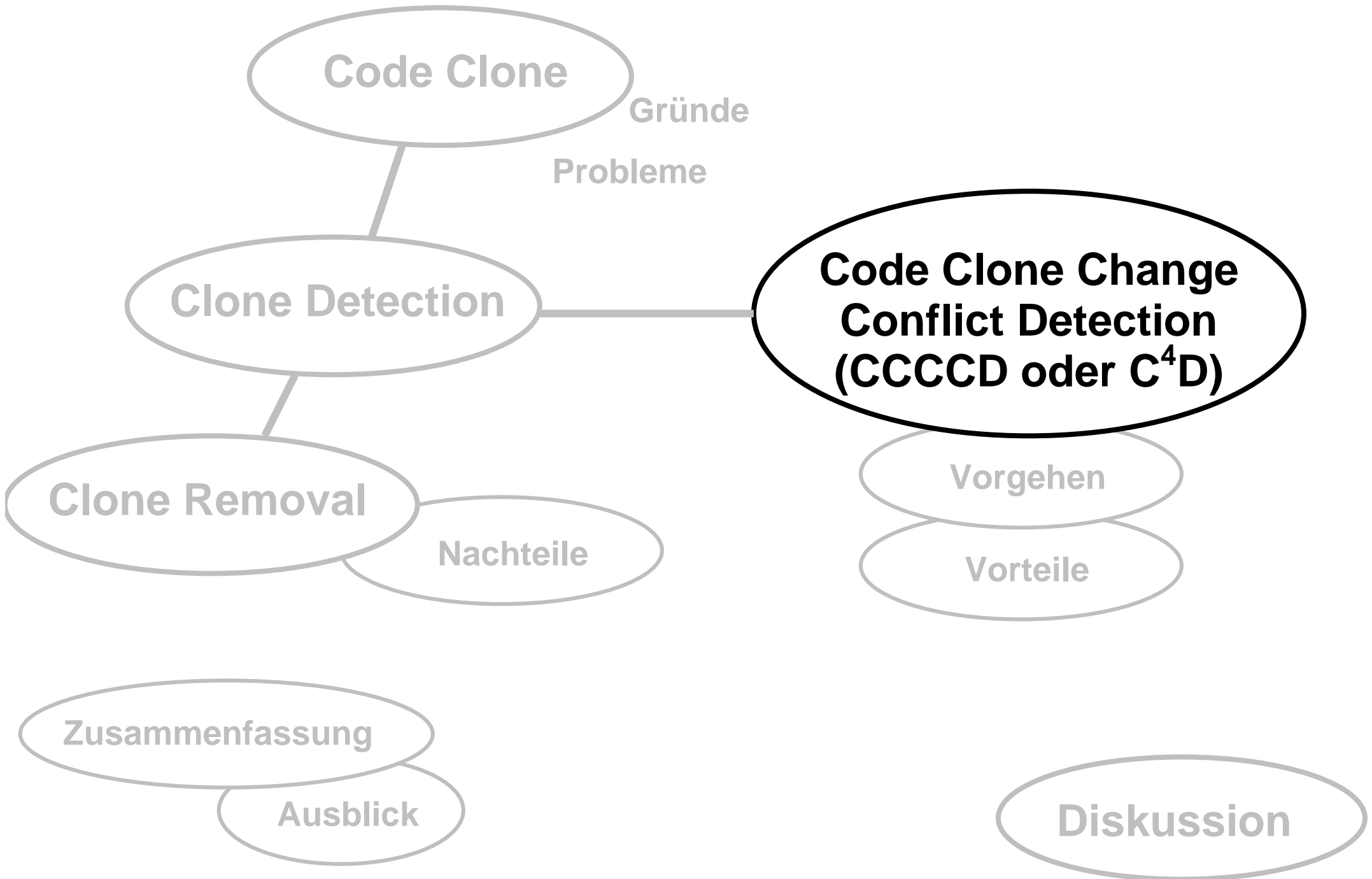


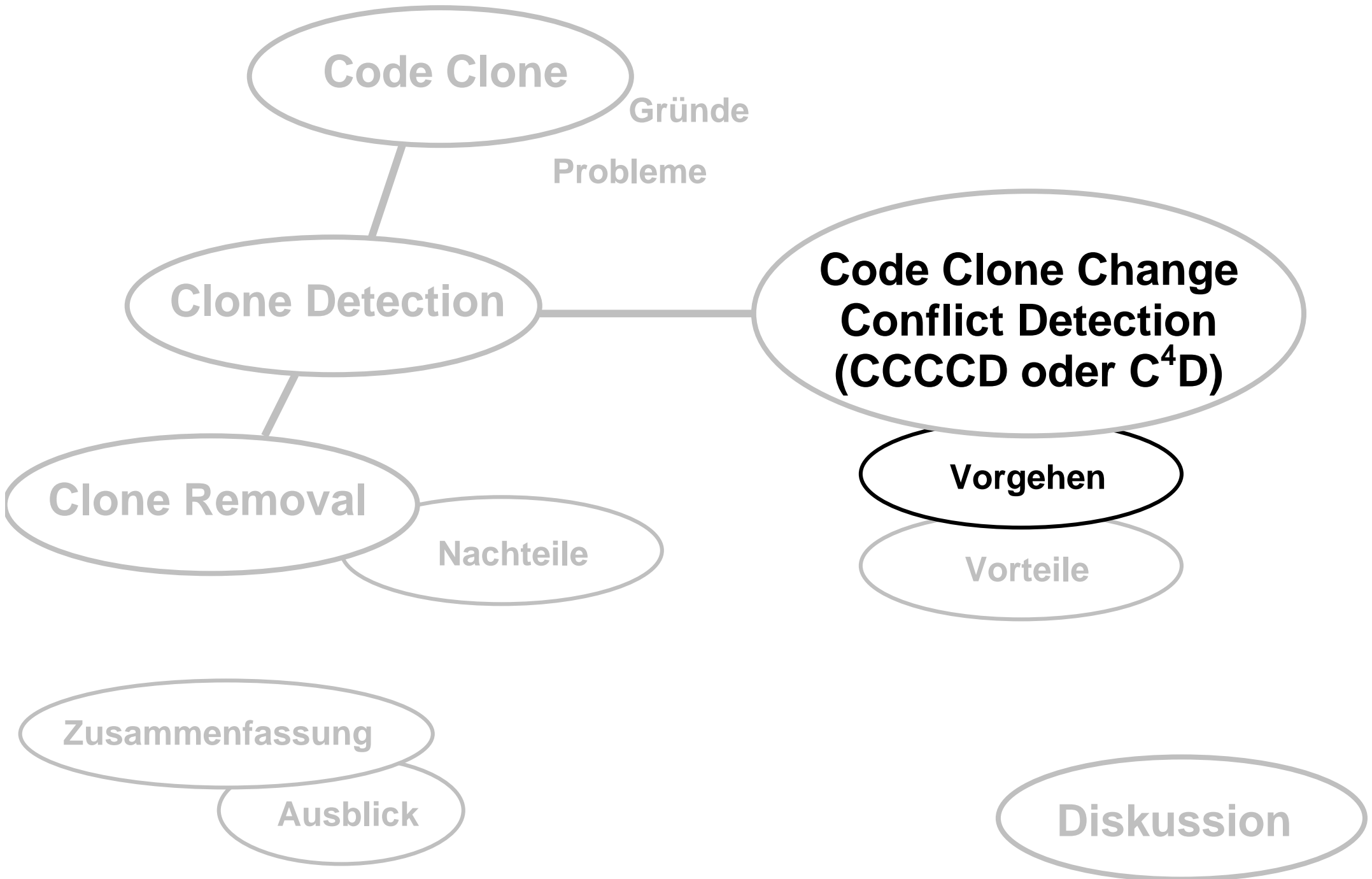




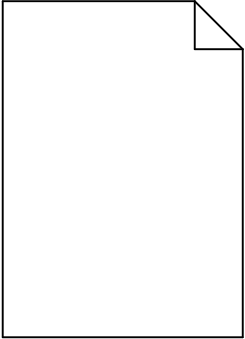




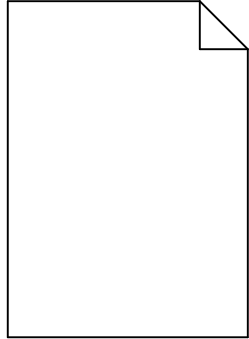




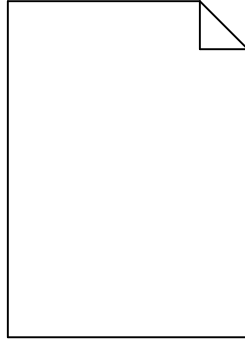
F1



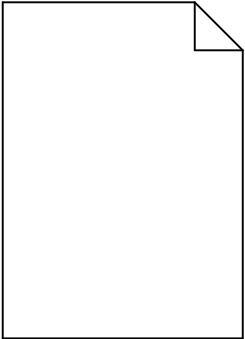
F2



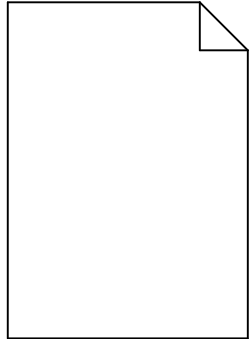
F3



F4

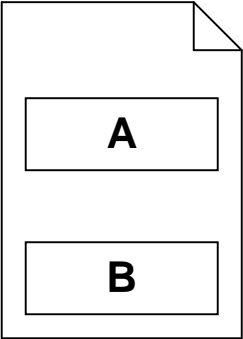


F5

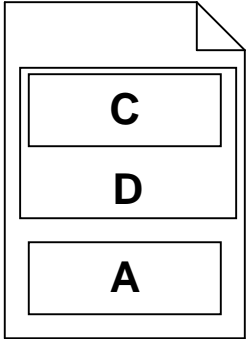


# Clone Detection

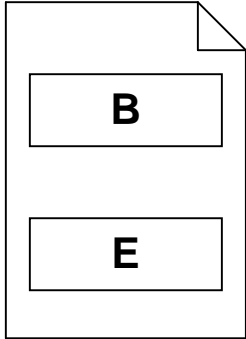
F1



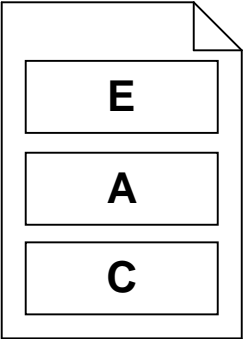
F2



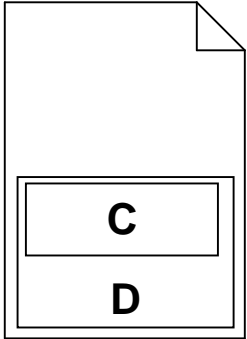
F3

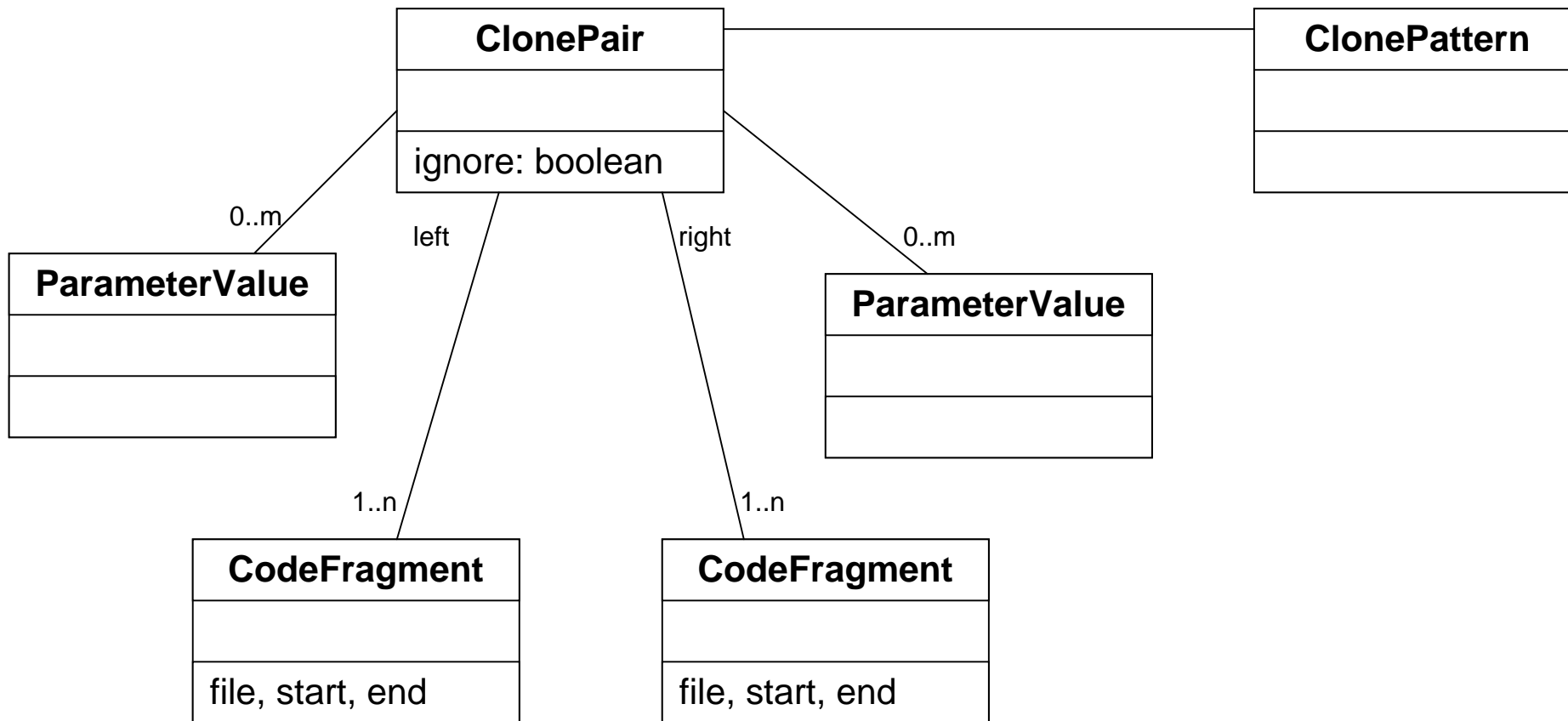


F4



F5





```

public static String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }

        File file = new File("../..//conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..//conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        Log.error\("ReportService",
            "Error finding engine.xml: " + ex.getMessage());
        return(null);
    }
}

```

```

public String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }
        ServletContext sc = getServletContext();
        String path =
            sc.getInitParameter("company.conf");
        if (path != null) {
            return(path);
        }
        File file = new File("../..//conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..//conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        System.err.println\(
            "Error finding engine.xml: " +
            ex.getMessage());
        return(null);
    }
}

```

```

public String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }
        File file = new File("../..../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        return("Error finding engine.xml: " + ex.getMessage());
    }
}

```

```

public static String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }
        ServletContext sc = getServletContext();
        String path =
            sc.getInitParameter("company.conf");
        if (path != null) {
            return(path);
        }
        File file = new File("../..../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        System.err.println(
            "Error finding engine.xml: " +
            ex.getMessage());
        return(null);
    }
}

```



## eclipse-jdtcore/src/.../ProblemReporter.java 1718 1734

### [\[Other part\]](#)

```
public void staticAndInstanceConflict(MethodBinding currentMethod, MethodBinding inheritedMethod) {
    if (currentMethod.isStatic\(\))
        this.handle(
            // This static method cannot hide the instance method from %1
            // 8.4.6.4 - If a class inherits more than one method with the same signature a
            // static (non-abstract) method cannot hide an instance method.
            CannotHideAnInstanceMethodWithAStaticMethod,
            new String[] {new String(inheritedMethod.declaringClass.readableName())},
            currentMethod.sourceStart(),
            currentMethod.sourceEnd());
    else
        this.handle(
            // This instance method cannot override the static method from %1
            // 8.4.6.4 - If a class inherits more than one method with the same signature
            // an instance (non-abstract) method cannot override a static method.
            CannotOverrideAStaticMethodWithAnInstanceMethod,
            new String[] {new String(inheritedMethod.declaringClass.readableName())},
            currentMethod.sourceStart(),
            currentMethod.sourceEnd()); }
```

## eclipse-jdtcore/src/.../ProblemReporter.java 840 873

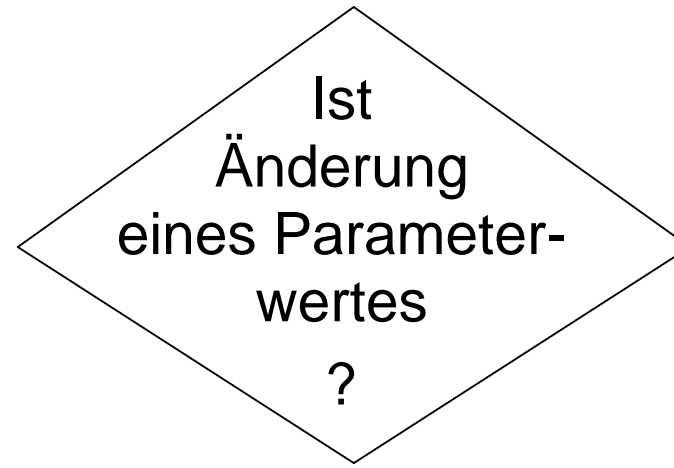
### [\[Other part\]](#)

```
public void incompatibleExceptionInThrowsClause(SourceTypeBinding type, MethodBinding currentMethod,  
                                                MethodBinding inheritedMethod, ReferenceBinding exceptionType) {  
    if (type == currentMethod.declaringClass)  
        this.handle(  
            // Exception %1 is not compatible with throws clause in %2  
            // 9.4.4 - The type of exception in the throws clause is incompatible.  
            IncompatibleExceptionInThrowsClause,  
            new String[] {  
                new String(exceptionType.sourceName\(\)),  
                new String(  
                    CharOperation.concat(  
                        inheritedMethod.declaringClass.readableName(),  
                        inheritedMethod.readableName\(\),  
                        '.'))},  
                currentMethod.sourceStart(),  
                currentMethod.sourceEnd());  
    else  
        this.handle(  
            // Exception %1 in throws clause of %2 is not compatible with %3  
            // 9.4.4 - The type of exception in the throws clause is incompatible.  
            IncompatibleExceptionInInheritedMethodThrowsClause,  
            new String[] {  
                new String(exceptionType.sourceName\(\)),  
                new String(  
                    CharOperation.concat(  
                        currentMethod.declaringClass.sourceName(),  
                        currentMethod.readableName(),  
                        '.')),  
                new String(  
                    CharOperation.concat(  
                        inheritedMethod.declaringClass.readableName(),  
                        inheritedMethod.readableName\(\),  
                        '.'))},  
                type.sourceStart\(\),  
                type.sourceEnd\(\)); }  
}
```

## Clone Pattern

```
public void ${p1}(${p2} MethodBinding currentMethod, MethodBinding inheritedMethod${p3}) {
    if (${p4} currentMethod.${p5}${p6})${p7}
        this.handle(
            // Exception %1 is not compatible with throws clause in %2
            // 9.4.4 - The type of exception in the throws clause is incompatible.
            ${p8},
            new String[] {
                new String(${p9}
                    inheritedMethod.declaringClass.readableName() ${p10})${p11}},
            currentMethod.sourceStart(),
            currentMethod.sourceEnd());
    else
        this.handle(
            // Exception %1 in throws clause of %2 is not compatible with %3
            // 9.4.4 - The type of exception in the throws clause is incompatible.
            ${p12},
            new String[] {
                new String(${p13}
                    inheritedMethod.declaringClass.readableName() ${p10})${p11}},
            ${p14}.sourceStart(),
            ${p14}.sourceEnd()); }
}
```

# Bei Änderung in Code-Fragment eines ClonePairs



```

public String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null)
        {
            return(System.getProperty("company.conf"));
        }
        ServletContext sc = getServletContext();
        String path =
            sc.getInitParameter("company.conf");
        if (path != null) {
            return(path);
        }
        File file = new File("../../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        System.err.println(
            "Error finding engine.xml: " +
            ex.getMessage());
        return(null);
    }
}

```

```

public String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null)
        {
            return(System.getProperty("company.conf"));
        }
        ServletContext sc = getServletContext();
        String path =
            sc.getInitParameter("MyConfigPath");
        if (path != null) {
            return(path);
        }
        File file = new File("../../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        System.err.println(
            "Error finding engine.xml: " +
            ex.getMessage());
        return(null);
    }
}

```

# Bei Änderung in Code-Fragment eines ClonePairs



```

public static String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }

        File file = new File("../..../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(System.getProperty("user.dir"));
        }
    } catch (Exception ex) {
        Log.error("ReportService",
            "Error finding engine.xml: " + ex.getMessage());
        return(null);
    }
}

```

```

public String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null) {
            return(System.getProperty("company.conf"));
        }
        ServletContext sc = getServletContext();
        String path =
            sc.getInitParameter("company.conf");
        if (path != null) {
            return(path);
        }
        File file = new File("../..../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../..../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(null);
        }
    } catch (Exception ex) {
        System.err.println(
            "Error finding engine.xml: " +
            ex.getMessage());
        return(null);
    }
}

```

```

public #{p1} String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null)
        {
            return(System.getProperty("company.conf"));
#{p2}
        }
        File file = new File("../../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(#{p4});
        }
    } catch (Exception ex) {
#{p3}
        "Error finding engine.xml: " +
ex.getMessage());
        return(null);
    }
}

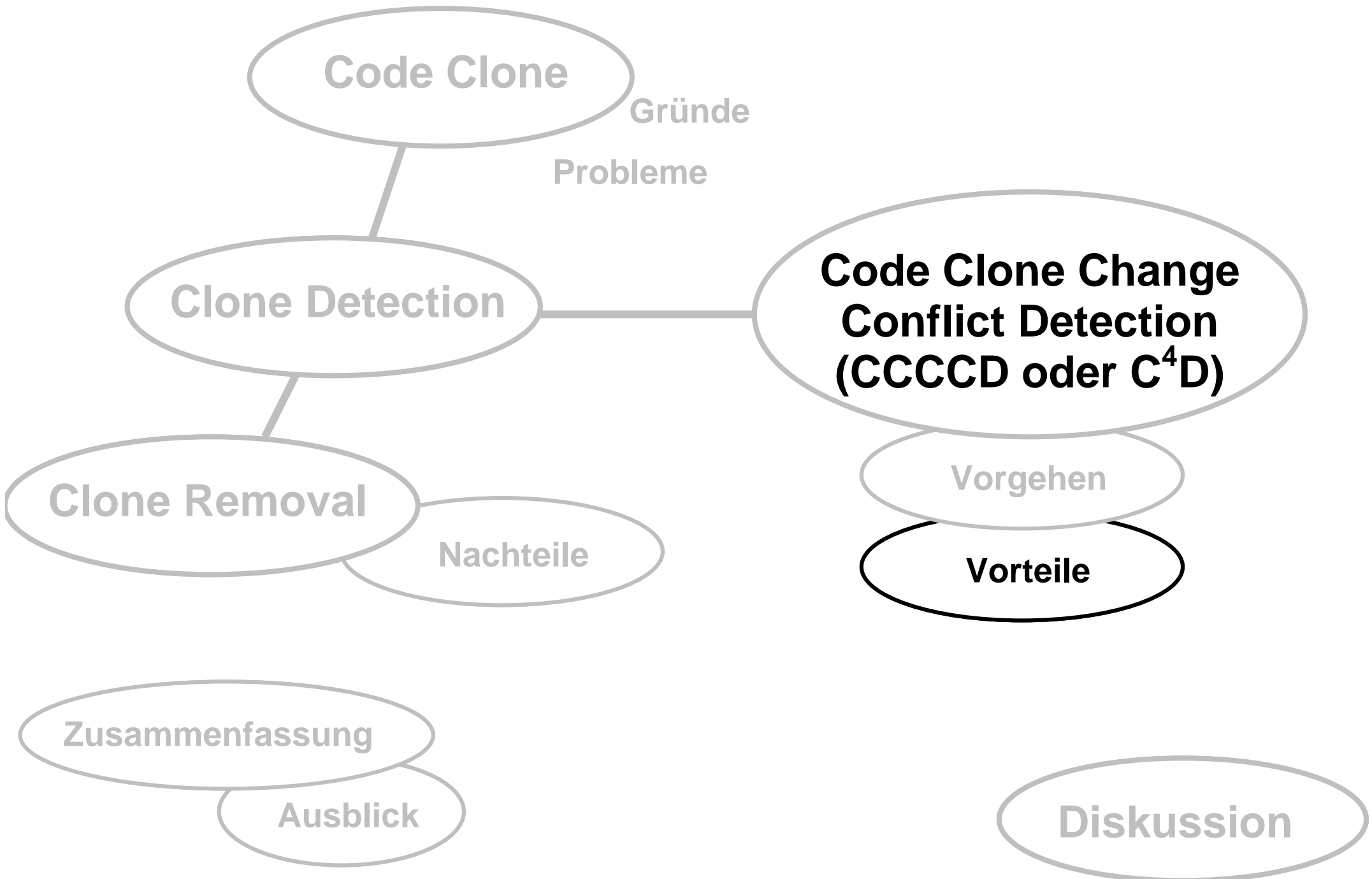
```

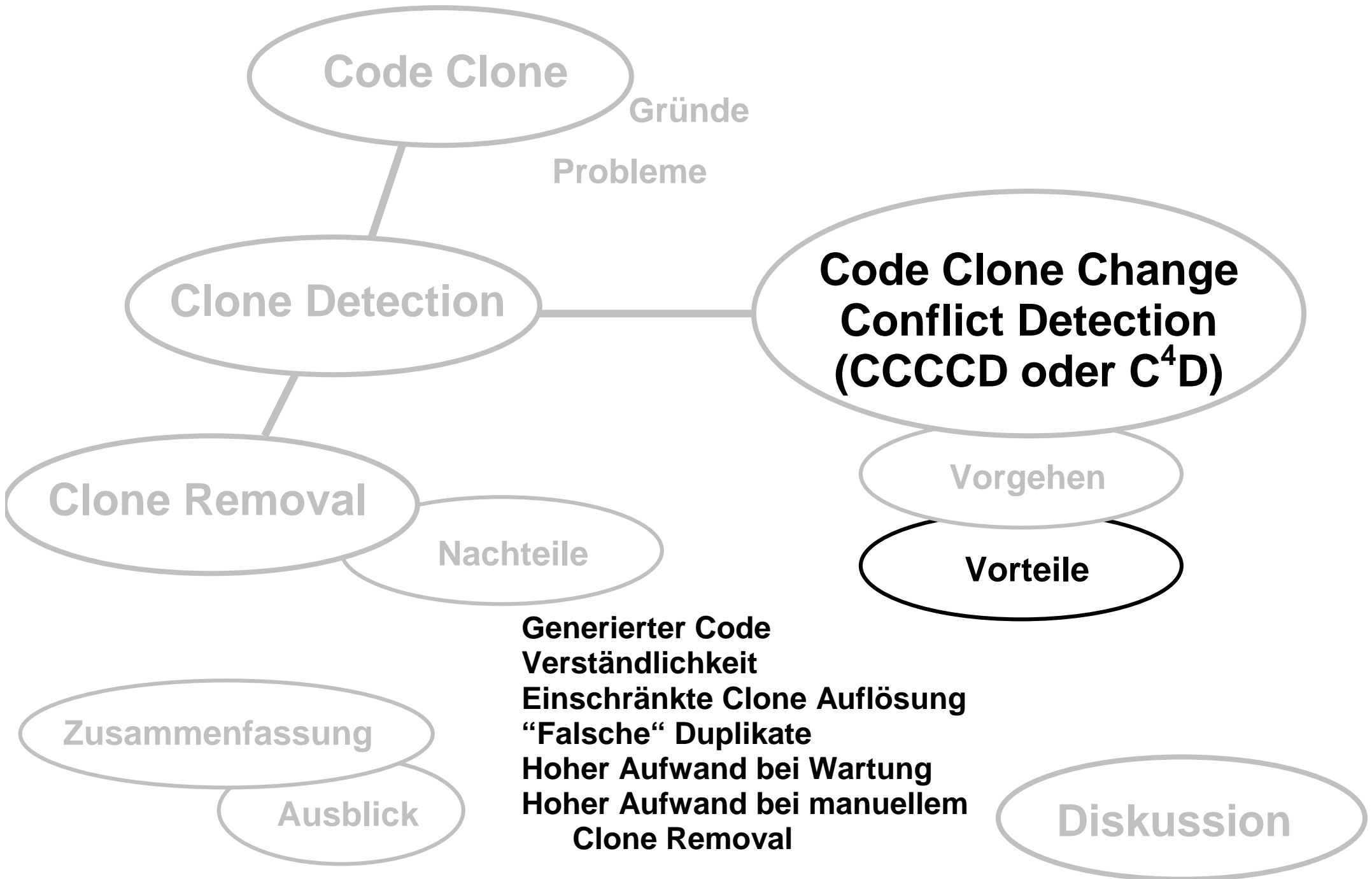
```

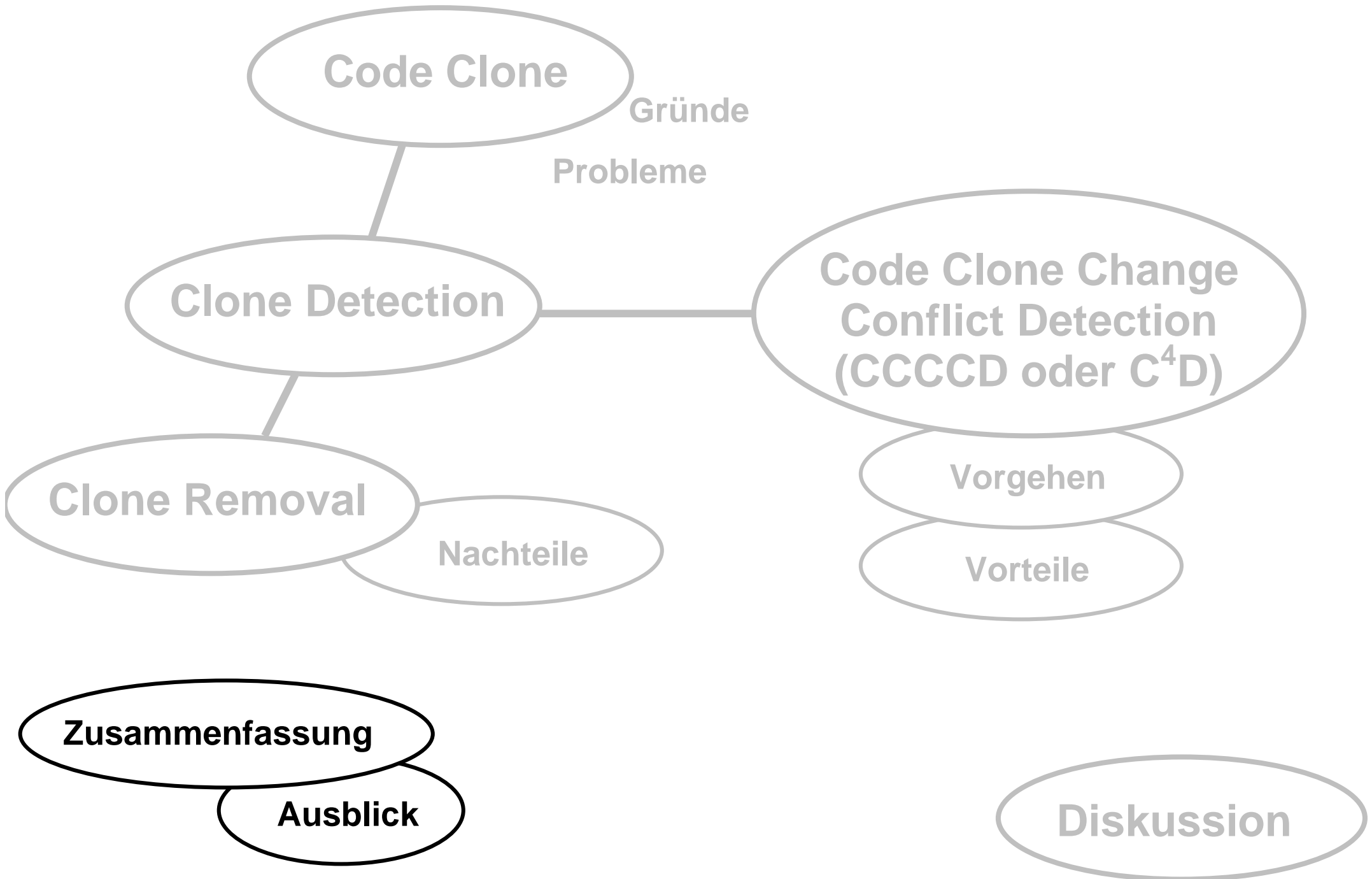
public #{p1} String getConfPath() {
    try {
        if (System.getProperty("company.conf") != null)
        {
            return(System.getProperty("company.conf"));
#{p2}
        }
        File file = new File("../../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../../conf");
        }
        file = new File("../conf/engine.xml");
        if (file != null && file.isFile()) {
            return("../conf");
        }
        file = new File("conf/engine.xml");
        if (file != null && file.isFile()) {
            return("conf");
        } else {
            return(System.getProperty("user.dir"));
        }
    } catch (Exception ex) {
#{p3}
        "Error finding engine.xml: " +
ex.getMessage());
        return(null);
    }
}

```









# Code Clone Change Conflict Detection (C<sup>4</sup>D)

## Diskussion

Website: [www.Udo-Borkowski.de/C4D](http://www.Udo-Borkowski.de/C4D)

