

Git



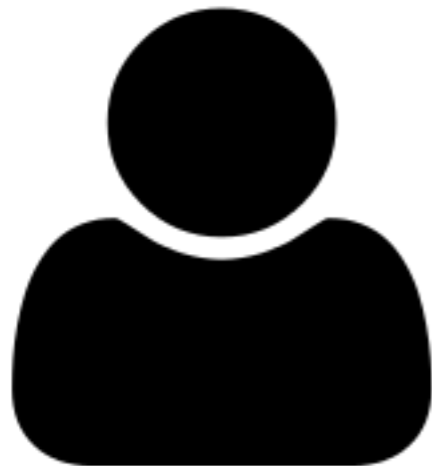
Productivity Tools for Developers

- Editors
- Debuggers
- Build automation tools
- Integrated development environment
- Modeling tools
- Source code analyzers / testing tools
- Version control tools

Single Developer



Single Developer



```
protected Collection<String> genAtomicPropositions(
    int size) {
    List<String> props = new ArrayList<String>();

    if (type == AlphabetType.PROPOSITIONAL) {
        int base = 15;
        for (int j = 0; j < size; j++) {
            char n = (char) ('a' + (base++ % 26));
            props.add(String.valueOf(n));
        }
    } else if (type == AlphabetType.CLASSICAL) {
        for (int j = 0; j < size; j++) {
            String sym = "a" + j;
            props.add(sym);
        }
    }

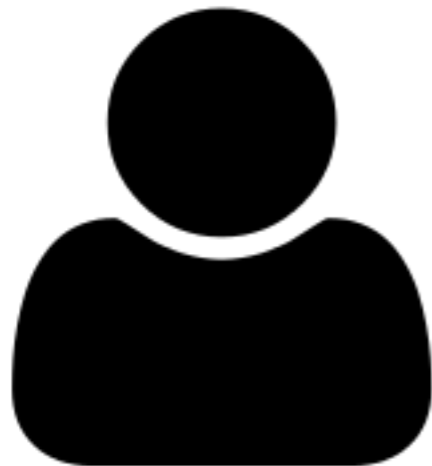
    return props;
}
```

```
protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    if (type == AlphabetType.PROPOSITIONAL) {
        int max = (int) Math.pow(2, props.size());
        for (int j = 0; j < max; j++) {
            String symbol = "";
            int n = j;
            for (String prop : props) {
                if (n % 2 == 0)
                    symbol += " " + prop;
                else
                    symbol += " ~" + prop;
                n /= 2;
            }
            alphabet.add(symbol.trim());
        }
    } else if (type == AlphabetType.CLASSICAL) {
        alphabet.addAll(props);
    }

    return alphabet;
}
```

Single Developer



```
protected Collection<String> genAtomicPropositional(
    int size) {
    List<String> props = new ArrayList<String>();

    if (type == AlphabetType.PROPOSITIONAL) {
        int base = 15;
        for (int j = 0; j < size; j++) {
            char n = (char) ('a' + (base++ % 26));
            props.add(String.valueOf(n));
        }
    } else if (type == AlphabetType.CLASSICAL) {
        for (int j = 0; j < size; j++) {
            String sym = "a" + j;
            props.add(sym);
        }
    }

    return props;
}
```

```
protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    if (type == AlphabetType.PROPOSITIONAL) {
        int max = (int) Math.pow(2, props.size());
        for (int j = 0; j < max; j++) {
            String symbol = "";
            int n = j;
            for (String prop : props) {
                if (n % 2 == 0)
                    symbol += " " + prop;
                else
                    symbol += " ~" + prop;
                n /= 2;
            }
            alphabet.add(symbol.trim());
        }
    } else if (type == AlphabetType.CLASSICAL) {
        alphabet.addAll(props);
    }

    return alphabet;
}
```

Single Developer

```
int x = insets.left;
int y = insets.top;

int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}

for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();

    int width = orient == HORIZONTAL ? pref.width : w;
    int height = orient == VERTICAL ? pref.height : h;

    if (width < min.width)
        width = min.width;
    if (height < min.height)
        height = min.height;

    c.setBounds(x, y, width, height);

    if (orient == HORIZONTAL)
```

```
Definition upd (x : var) (v : value) (s : t) :=
  fun (y : var) => if y == x then v else acc y s.

Definition upd2 x1 v1 x2 v2 (s : t) : t :=
  upd x2 v2 (upd x1 v1 s).

Lemma acc_upd_eq x y v s :
  x == y ->
  acc x (upd y v s) = v.
Proof.
  rewrite /acc /upd => Hxy.
  rewrite Hxy.
  reflexivity.
Qed.

Lemma acc_upd_neq x y v s :
  x != y ->
  acc x (upd y v s) = acc x s.
Proof.
  rewrite {1}/acc /upd => Hxy.
  rewrite (negPf Hxy).
  reflexivity.
Qed.

Lemma acc_upd2_eq1 :
  forall x y1 v1 y2 v2 (s : t),
  x == y1 ->
  x != y2 ->
  acc x (upd2 y1 v1 y2 v2 s) = v1.
Proof.
  move=> x y1 v1 y2 v2 s Hx1 Hx2.
  rewrite /upd2 (acc_upd_neq _ _ Hx2) (acc_upd_eq _ _ Hx1).
  reflexivity.
```

```
opertyAsInteger(O_MAX_RETRY);

betType.fromString(options
  PHABET_TYPE));
enerationModel.valueOf(options
  NERATION_MODEL));
etPropertyAsInteger(O_PROPOSITION_SIZE);
opertyAsInteger(O_N_SIZE);
opertyAsInteger(O_D_SIZE);
PropertyAsDouble(O_N_TRANSITION_PROBABILITY);
PropertyAsDouble(O_N_SYMBOL_PROBABILITY);
PropertyAsDouble(O_N_TRANSITION_DENSITY);

ble(O_CONNECTING_SYMBOL_PROBABILITY);
PropertyAsDouble(O_D_ACCEPTANCE_PROBABILITY);
PropertyAsDouble(O_D_ACCEPTANCE_DENSITY);

= genAtomicPropositions(type, prop_size);
phabet(type, props).toArray(new String[0]);
```

```
protected Collection<String> genAtomicPropositions(
  int size) {
  List<String> props = new ArrayList<String>(size);

  if (type == AlphabetType.PROPOSITIONAL) {
    int base = 15;
    for (int j = 0; j < size; j++) {
      char n = (char) ('a' + (base++ % 26));
      props.add(String.valueOf(n));
    }
  } else if (type == AlphabetType.CLASSICAL) {
    for (int j = 0; j < size; j++) {
      String sym = "a" + j;
      props.add(sym);
    }
  }

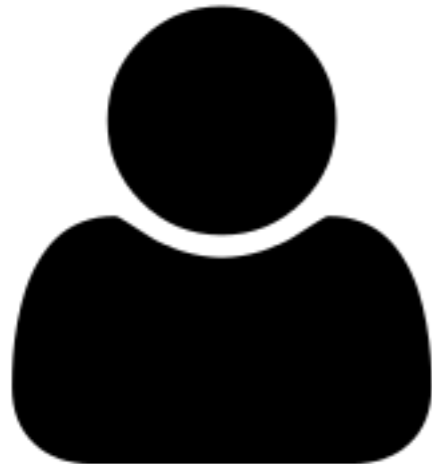
  return props;
}

String symbol =
  int n = j;
  for (String prop : props) {
    if (n % 2 == 0)
      symbol += prop;
    else
      symbol += prop + "\n";
    n /= 2;
  }
  alphabet.add(symbol);
} else if (type == AlphabetType.CLASSICAL) {
  alphabet.addAll(props);
}

return alphabet;
}
```

```
aut = new FSA(type, Position.OnTransition);
aut.expandAlphabet(props.toArray(new String[0]));

/* Generate states. */
State[] ustates = genStates(aut, useize);
State[] lstates = genStates(aut, lsize);
for (State s : ustates)
  s.setDescription("N");
for (State s : lstates)
  s.setDescription("D");
```



```
protected Collection<String> genAtomicPropositions(
  int size) {
  List<String> props = new ArrayList<String>(size);

  if (type == AlphabetType.PROPOSITIONAL) {
    int base = 15;
    for (int j = 0; j < size; j++) {
      char n = (char) ('a' + (base++ % 26));
      props.add(String.valueOf(n));
    }
  } else if (type == AlphabetType.CLASSICAL) {
    for (int j = 0; j < size; j++) {
      String sym = "a" + j;
      props.add(sym);
    }
  }

  return props;
}
}
```

Single Developer

```
int x = insets.left;
int y = insets.top;

int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}

for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();

    int width = orient == HORIZONTAL ? pref.width : w;
    int height = orient == VERTICAL ? pref.height : h;

    if (width < min.width)
        width = min.width;
    if (height < min.height)
        height = min.height;

    c.setBounds(x, y, width, height);

    if (orient == HORIZONTAL)
```

```
Definition upd (x : var) (v : value) (s : t) :=
  fun (y : var) => if y == x then v else acc y s.

Definition upd2 x1 v1 x2 v2 (s : t) : t :=
  upd x2 v2 (upd x1 v1 s).

Lemma acc_upd_eq x y v s :
  x == y ->
  acc x (upd y v s) = v.
Proof.
  rewrite /acc /upd => Hxy.
  rewrite Hxy.
  reflexivity.
Qed.

Lemma acc_upd_neq x y v s :
  x != y ->
  acc x (upd y v s) = acc x s.
Proof.
  rewrite {1}/acc /upd => Hxy.
  rewrite (negPf Hxy).
  reflexivity.
Qed.

Lemma acc_upd2_eq1 :
  forall x y1 v1 y2 v2 (s : t),
  x == y1 ->
  x != y2 ->
  acc x (upd2 y1 v1 y2 v2 s) = v1.
Proof.
  move=> x y1 v1 y2 v2 s Hx1 Hx2.
  rewrite /upd2 (acc_upd_neq _ _ Hx2) (acc_upd_eq _ _ Hx1).
  reflexivity.
```

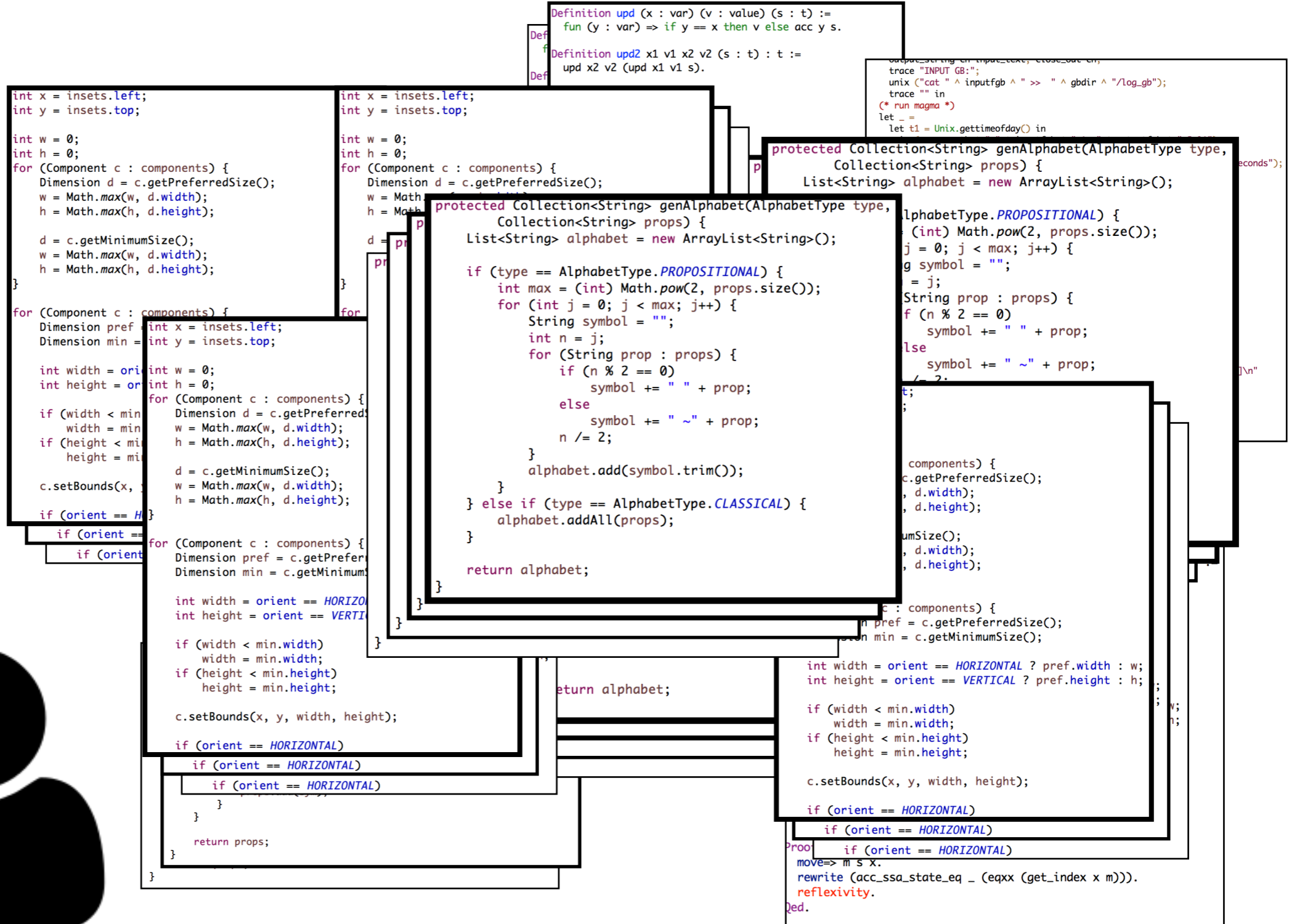
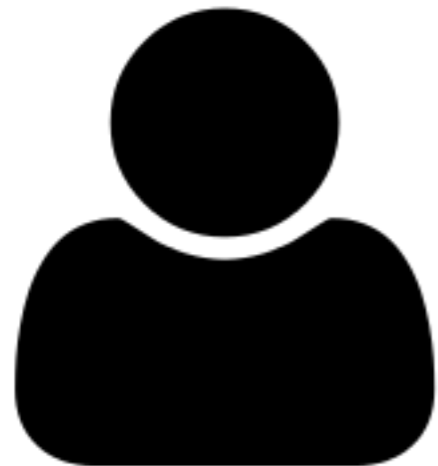
```
protected Collection<String> genAtomicPropositionalAlphabet (type,
  List<String> props = new ArrayList<String>(
  if (type == AlphabetType.PROPOSITIONAL) {
    int base = 15;
    for (int j = 0; j < size; j++) {
      char n = (char) ('a' + (base++ % 26));
      props.add(String.valueOf(n));
    }
  } else if (type == AlphabetType.CLASSICAL) {
    for (int j = 0; j < size; j++) {
      String sym = "a" + j;
      props.add(sym);
    }
  }
  return props;
}

String symbol = "";
int n = j;
for (String prop : props) {
  if (n % 2 == 0)
    symbol += " " + prop;
  else
    symbol += "~" + prop;
  n /= 2;
}
alphabet.add(symbol.trim());
} else if (type == AlphabetType.CLASSICAL) {
  alphabet.addAll(props);
}
return alphabet;
```

```
protected Collection<String> genAtomicPropositionalAlphabet (
  int size) {
  List<String> props = new ArrayList<String>(
  if (type == AlphabetType.PROPOSITIONAL) {
    int base = 15;
    for (int j = 0; j < size; j++) {
      char n = (char) ('a' + (base++ % 26));
      props.add(String.valueOf(n));
    }
  } else if (type == AlphabetType.CLASSICAL) {
    for (int j = 0; j < size; j++) {
      String sym = "a" + j;
      props.add(sym);
    }
  }
  return props;
}
```

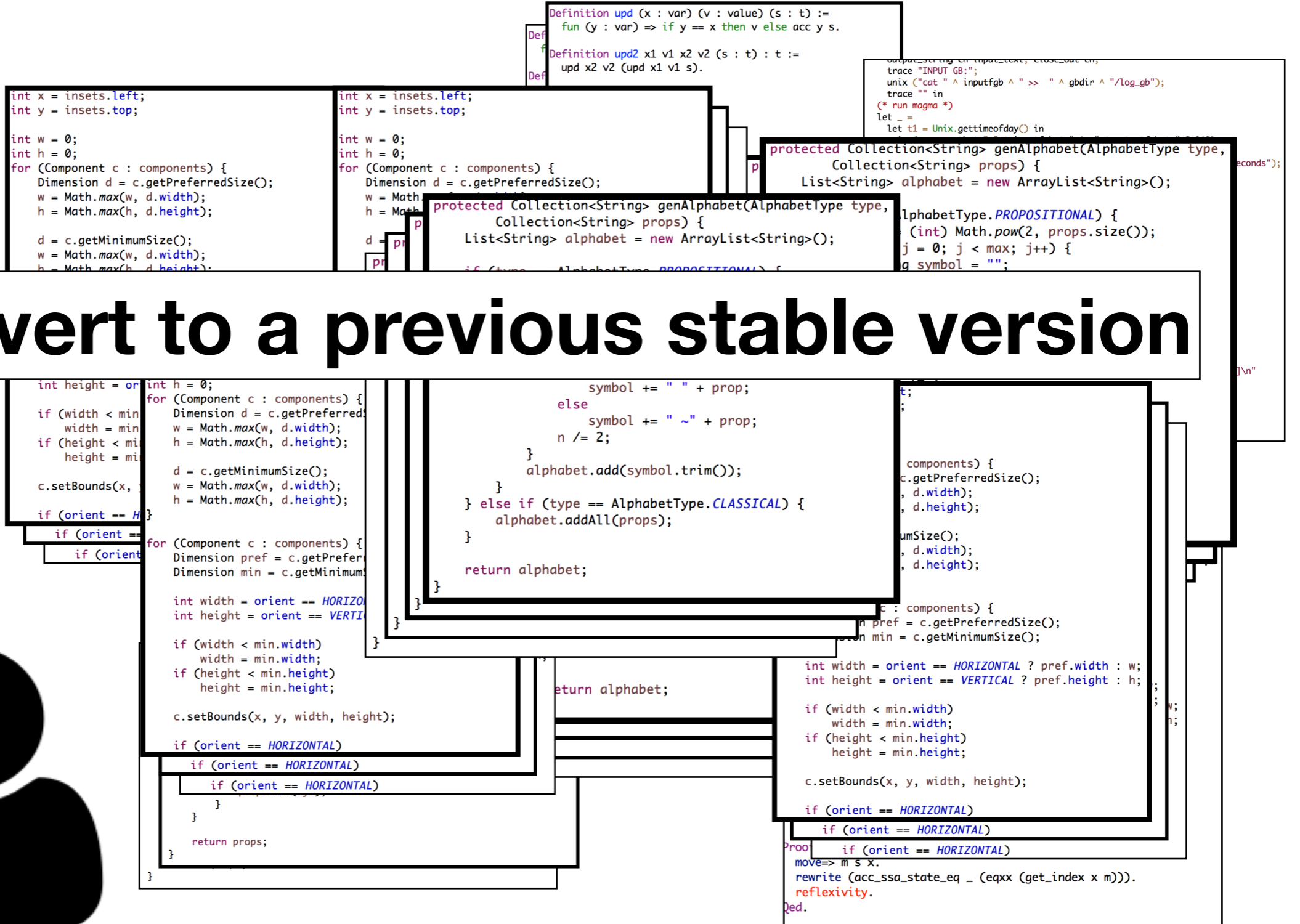


Single Developer



Single Developer

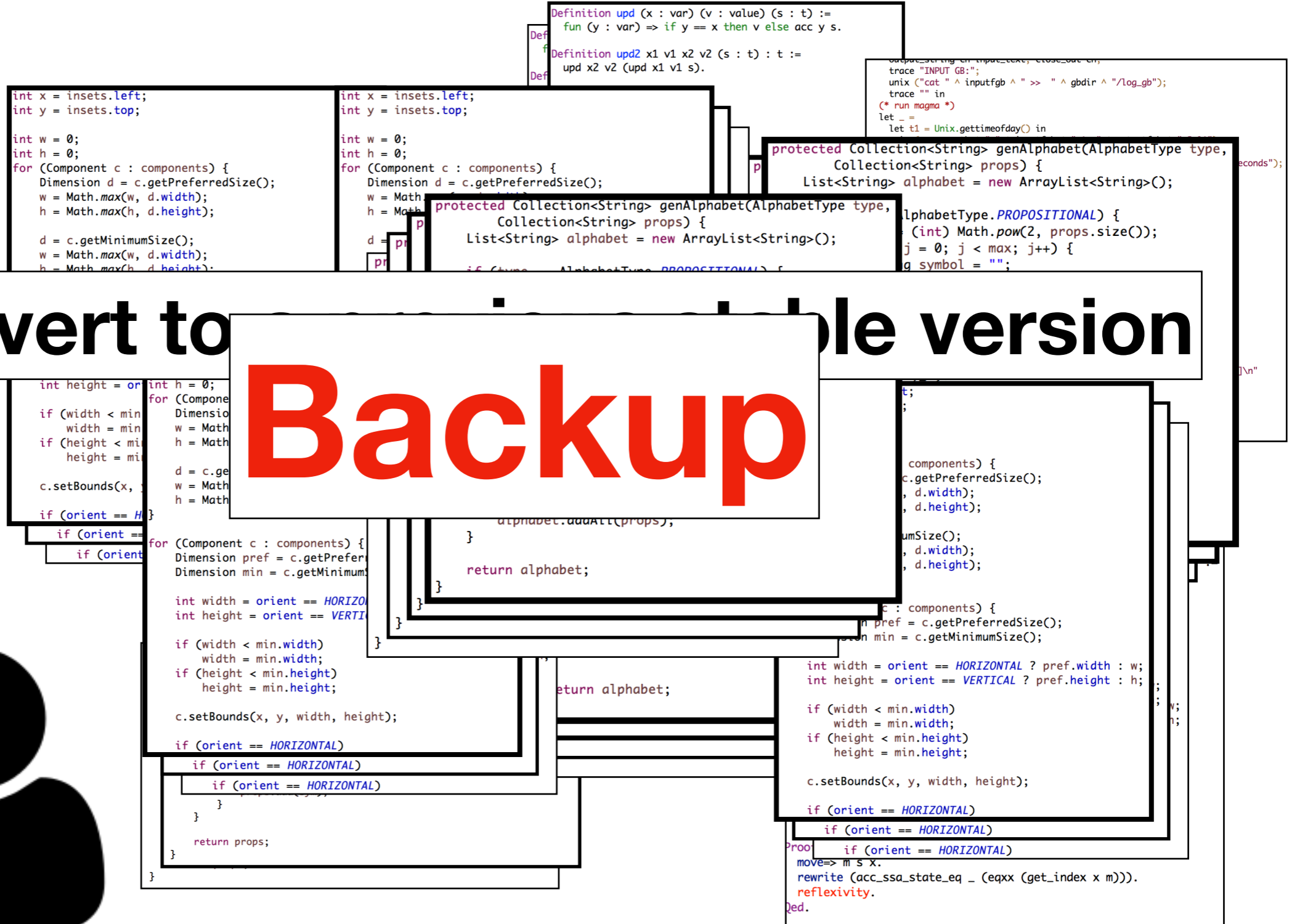
Revert to a previous stable version



Single Developer

Revert to previous stable version

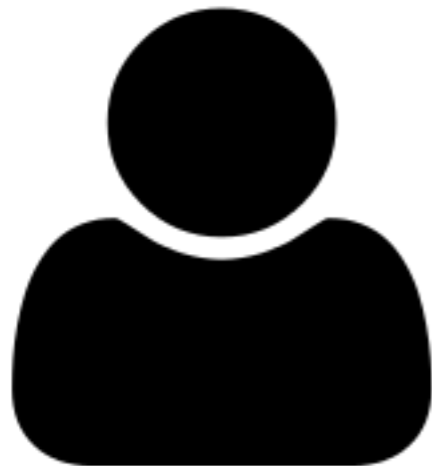
Backup



Project Backup



Project Backup



```
protected Collection<String> genAtomicPropositions(int size) {
    List<String> props = new ArrayList<String>(size);
    if (type == AlphabetType.PROPOSITIONAL) {
        int base = 15;
        for (int j = 0; j < size; j++) {
            char n = (char) ('a' + (base++ % 26));
            props.add(String.valueOf(n));
        }
    } else if (type == AlphabetType.CLASSICAL) {
        for (int j = 0; j < size; j++) {
            String sym = "a" + j;
            props.add(sym);
        }
    }
    return props;
}

protected Collection<String> genAlphabet(AlphabetType type, int prop_size, int use_size, int lsize, double N_pt, double N_ps, double N_dt, double conn_ps, double D_pa, double D_da) {
    Collection<String> props = genAtomicPropositions(type, prop_size);
    String[] alphabet = genAlphabet(type, props).toArray(new String[0]);

    FSA aut = null;
    boolean done = false;
    while (!done) {
        aut = new FSA(type, Position.OnTransition);
        aut.expandAlphabet(props.toArray(new String[0]));

        /* Generate states. */
        State[] ustates = genStates(aut, use_size);
        State[] lstates = genStates(aut, lsize);
        for (State s : ustates)
            s.setDescription("N");
        for (State s : lstates)
            s.setDescription("D");
    }

    if (type == AlphabetType.CLASSICAL) {
        alphabet.addAll(props);
    }

    return alphabet;
}

int retry = options.getPropertyAsInteger(O_MAX_RETRY);

AlphabetType type = AlphabetType.fromString(options.getProperty(O_ALPHABET_TYPE));
GenerationModel model = GenerationModel.valueOf(options.getProperty(O_GENERATION_MODEL));
int prop_size = options.getPropertyAsInteger(O_PROPOSITION_SIZE);
int use_size = options.getPropertyAsInteger(O_N_SIZE);
int lsize = options.getPropertyAsInteger(O_D_SIZE);
double N_pt = options.getPropertyAsDouble(O_N_TRANSITION_PROBABILITY);
double N_ps = options.getPropertyAsDouble(O_N_SYMBOL_PROBABILITY);
double N_dt = options.getPropertyAsDouble(O_N_TRANSITION_DENSITY);
double conn_ps = options.getPropertyAsDouble(O_CONNECTING_SYMBOL_PROBABILITY);
double D_pa = options.getPropertyAsDouble(O_D_ACCEPTANCE_PROBABILITY);
double D_da = options.getPropertyAsDouble(O_D_ACCEPTANCE_DENSITY);

Collection<String> props = genAtomicPropositions(type, prop_size);
String[] alphabet = genAlphabet(type, props).toArray(new String[0]);

FSA aut = null;
boolean done = false;
while (!done) {
    aut = new FSA(type, Position.OnTransition);
    aut.expandAlphabet(props.toArray(new String[0]));

    /* Generate states. */
    State[] ustates = genStates(aut, use_size);
    State[] lstates = genStates(aut, lsize);
    for (State s : ustates)
        s.setDescription("N");
    for (State s : lstates)
        s.setDescription("D");
}

if (type == AlphabetType.CLASSICAL) {
    alphabet.addAll(props);
}

return alphabet;
}
```

Project Backup



2017/08/01



```
int retry = options.getPropertyAsInteger(O_MAX_RETRY);

AlphabetType type = AlphabetType.fromString(options
    .getProperty(O_ALPHABET_TYPE));
GenerationModel model = GenerationModel.valueOf(options
    .getProperty(O_GENERATION_MODEL));
int prop_size = options.getPropertyAsInteger(O_PROPOSITION_SIZE);
int usize = options.getPropertyAsInteger(O_N_SIZE);
int lsize = options.getPropertyAsInteger(O_D_SIZE);
double N_pt = options.getPropertyAsDouble(O_N_TRANSITION_PROBABILITY);
double N_ps = options.getPropertyAsDouble(O_N_SYMBOL_PROBABILITY);
double N_dt = options.getPropertyAsDouble(O_N_TRANSITION_DENSITY);
double conn_ps = options
    .getPropertyAsDouble(O_CONNECTING_SYMBOL_PROBABILITY);
double D_pa = options.getPropertyAsDouble(O_D_ACCEPTANCE_PROBABILITY);
double D_da = options.getPropertyAsDouble(O_D_ACCEPTANCE_DENSITY);

Collection<String> props = genAtomicPropositions(type, prop_size);
String[] alphabet = genAlphabet(type, props).toArray(new String[0]);

FSA aut = null;
boolean done = false;
while (!done) {
    aut = new FSA(type, Position.OnTransition);
    aut.expandAlphabet(props.toArray(new String[0]));

    /* Generate states. */
    State[] ustates = genStates(aut, usize);
    State[] lstates = genStates(aut, lsize);
    for (State s : ustates)
        s.setDescription("N");
    for (State s : lstates)
        s.setDescription("D");
}

return alphabet;
} else if (type == AlphabetType.CLASSICAL) {
    alphabet.addAll(props);
}

return alphabet;
}

protected Collection<String> genAtomicPropositions(
    int size) {
    List<String> props = new ArrayList<String>(size);

    if (type == AlphabetType.PROPOSITIONAL) {
        int base = 15;
        for (int j = 0; j < size; j++) {
            char n = (char) ('a' + (base++ % 26));
            props.add(String.valueOf(n));
        }
    } else if (type == AlphabetType.CLASSICAL) {
        for (int j = 0; j < size; j++) {
            String sym = "a" + j;
            props.add(sym);
        }
    }

    return props;
}

protected Collection<String> genAlphabet(
    AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>(props.size());

    if (type == AlphabetType.PROPOSITIONAL) {
        int max = (int) Math.ceil(Math.log(props.size()) / Math.log(2));
        for (int j = 0; j < max; j++) {
            String symbol = "";
            int n = j;
            for (String prop : props) {
                if (n % 2 == 0)
                    symbol += prop;
                else
                    symbol += prop.charAt(0);
                n /= 2;
            }
            alphabet.add(symbol);
        }
    } else if (type == AlphabetType.CLASSICAL) {
        alphabet.addAll(props);
    }

    return alphabet;
}
}
```

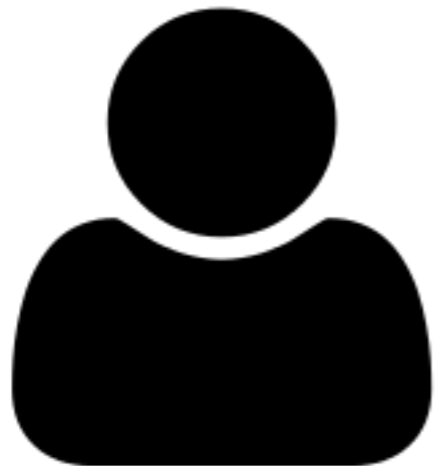
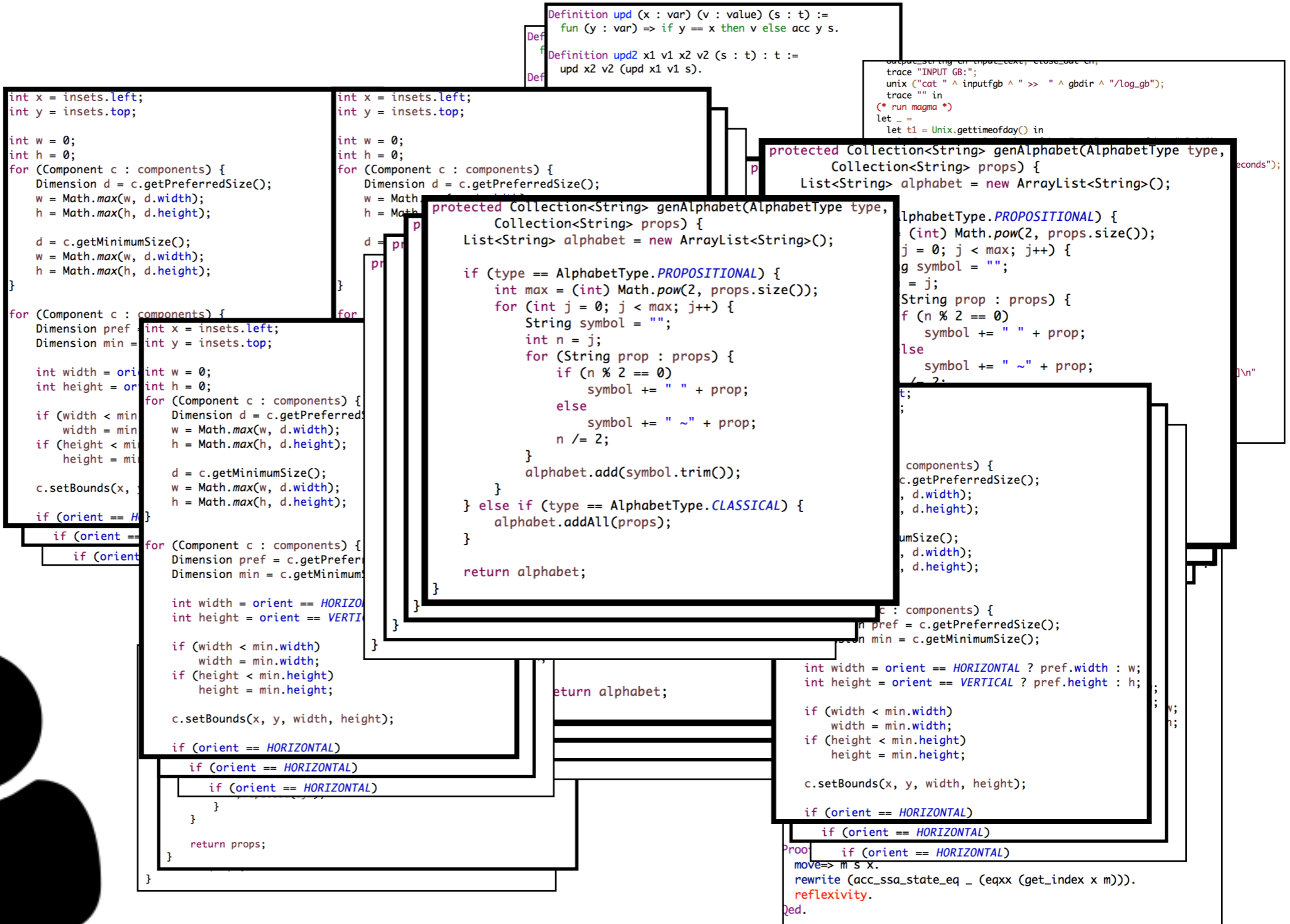
Project Backup

2017/08/01

2017/08/05

2017/08/13

2017/08/27



Project Backup

What has been done in a version?

2017

2017/08/05

```
int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}
```

2017/08/13

```
for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();

    int width = orient == HORIZONTAL ? pref.width : min.width;
    int height = orient == VERTICAL ? pref.height : min.height;

    c.setBounds(x, y, width, height);

    if (orient == HORIZONTAL)
        if (orient == HORIZONTAL)
            if (orient == HORIZONTAL)
                return props;
            }
        }
    }
}
```

2017/08/27

```
for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();

    int width = orient == HORIZONTAL ? pref.width : min.width;
    int height = orient == VERTICAL ? pref.height : min.height;

    if (width < min.width)
        width = min.width;
    if (height < min.height)
        height = min.height;

    c.setBounds(x, y, width, height);

    if (orient == HORIZONTAL)
        if (orient == HORIZONTAL)
            if (orient == HORIZONTAL)
                return props;
            }
        }
    }
}
```

```
int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}
```

```
protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    if (type == AlphabetType.PROPOSITIONAL) {
        int max = (int) Math.pow(2, props.size());
        for (int j = 0; j < max; j++) {
            String symbol = "";
            int n = j;
            for (String prop : props) {
                if (n % 2 == 0)
                    symbol += " " + prop;
                else
                    symbol += "~" + prop;
                n /= 2;
            }
            alphabet.add(symbol.trim());
        }
    } else if (type == AlphabetType.CLASSICAL) {
        alphabet.addAll(props);
    }

    return alphabet;
}
```

```
protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    if (type == AlphabetType.PROPOSITIONAL) {
        int max = (int) Math.pow(2, props.size());
        for (int j = 0; j < max; j++) {
            String symbol = "";
            int n = j;
            for (String prop : props) {
                if (n % 2 == 0)
                    symbol += " " + prop;
                else
                    symbol += "~" + prop;
                n /= 2;
            }
            alphabet.add(symbol.trim());
        }
    } else if (type == AlphabetType.CLASSICAL) {
        alphabet.addAll(props);
    }

    return alphabet;
}
```

```
int width = orient == HORIZONTAL ? pref.width : w;
int height = orient == VERTICAL ? pref.height : h;

if (width < min.width)
    width = min.width;
if (height < min.height)
    height = min.height;

c.setBounds(x, y, width, height);

if (orient == HORIZONTAL)
    if (orient == HORIZONTAL)
        if (orient == HORIZONTAL)
            return props;
        }
    }
}
```



Project Backup

2017/08/05
2017/08/13
2017/08/27

What has been changed in a version?

Log

```
int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}

int width = orient == HORIZONTAL ? w : min.width;
int height = orient == VERTICAL ? h : min.height;

c.setBounds(x, y, width, height);

if (orient == HORIZONTAL)
    if (orient == HORIZONTAL)
        return props;
}
```

```
protected Collection<String> genAlphabet(AlphabetType type,
Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    genAlphabet(AlphabetType type,
    props) {
        new ArrayList<String>();

        if (type == AlphabetType.PROPOSITIONAL) {
            int max = (int) Math.pow(2, props.size());
            for (int j = 0; j < max; j++) {
                String symbol = "";
                int n = j;
                for (String prop : props) {
                    if (n % 2 == 0)
                        symbol += " " + prop;
                    else
                        symbol += "~" + prop;
                    n /= 2;
                }
                alphabet.add(symbol.trim());
            }
        } else if (type == AlphabetType.CLASSICAL) {
            alphabet.addAll(props);
        }

        return alphabet;
    }
}
```

```
let t1 = Unix.gettimeofday() in
protected Collection<String> genAlphabet(AlphabetType type,
Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    genAlphabet(AlphabetType type,
    props) {
        new ArrayList<String>();

        if (type == AlphabetType.PROPOSITIONAL) {
            (int) Math.pow(2, props.size());
            j = 0; j < max; j++) {
                String prop : props) {
                    if (n % 2 == 0)
                        symbol += " " + prop;
                    else
                        symbol += "~" + prop;
                    n /= 2;
                }
                alphabet.add(symbol.trim());
            }
        } else if (type == AlphabetType.CLASSICAL) {
            alphabet.addAll(props);
        }

        return alphabet;
    }
}
```



Project Backup

2017/08/05 **What has been changed in a version?**

Log

```
int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}
for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();
    int x = insets.left;
    int y = insets.top;
```

```
protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    genAlphabet(AlphabetType type,
    props) {
    new ArrayList<String>();
    AlphabetType.PROPOSITIONAL) {
    (int) Math.pow(2, props.size());
    j = 0; j < max; j++) {
    g symbol = "";
    = j;
    String prop : props) {
    f (n % 2 == 0)
    symbol += " " + prop;
    lse
```

How is a bug in version A fixed in version B?



```
c.setBounds(x, y, w, h);
if (orient == HORIZONTAL)
    if (orient == VERTICAL)
        if (orient == HORIZONTAL)
            if (orient == HORIZONTAL)
                return props;
            }
        }
    }
}

d = c.getMinimumSize();
w = Math.max(w, d.width);
h = Math.max(h, d.height);

for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();

    int width = orient == HORIZONTAL ? pref.width : w;
    int height = orient == VERTICAL ? pref.height : h;

    if (width < min.width)
        width = min.width;
    if (height < min.height)
        height = min.height;

    c.setBounds(x, y, width, height);

    if (orient == HORIZONTAL)
        if (orient == HORIZONTAL)
            return props;
        }
    }
}

alphabet.add(symbol.trim());
} else if (type == AlphabetType.CLASSICAL) {
    alphabet.addAll(props);
}
return alphabet;

return alphabet;

int width = orient == HORIZONTAL ? pref.width : w;
int height = orient == VERTICAL ? pref.height : h;

if (width < min.width)
    width = min.width;
if (height < min.height)
    height = min.height;

c.setBounds(x, y, width, height);

if (orient == HORIZONTAL)
    if (orient == HORIZONTAL)
        if (orient == HORIZONTAL)
            rewrite (acc_ssa_state_eq _ (eqxx (get_index x m))).
            reflexivity.
            ded.
```



Project Backup

2017/08/05 **What has been changed in a version?**

Log

```
int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}
for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();
    int x = insets.left;
    int y = insets.top;
```

```
protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    genAlphabet(AlphabetType type,
    props) {
    new ArrayList<String>();
    AlphabetType.PROPOSITIONAL) {
    (int) Math.pow(2, props.size());
    j = 0; j < max; j++) {
    g symbol = "";
    = j;
    String prop : props) {
    f (n % 2 == 0)
    symbol += " " + prop;
    lse
```

2017/08/05

2017/08/13

How is a bug in version A fixed in version B?

Diff

```
c.setBounds(x, y, w, h);
if (orient == HORIZONTAL)
    if (orient == VERTICAL)
        if (orient == HORIZONTAL)
            if (orient == VERTICAL)
                for (Component c : components)
                    Dimension pref = c.getPreferredSize();
                    Dimension min = c.getMinimumSize();

                    int width = orient == HORIZONTAL ? pref.width : w;
                    int height = orient == VERTICAL ? pref.height : h;

                    if (width < min.width)
                        width = min.width;
                    if (height < min.height)
                        height = min.height;

                    c.setBounds(x, y, width, height);

                    if (orient == HORIZONTAL)
                        if (orient == VERTICAL)
                            if (orient == HORIZONTAL)
                                if (orient == VERTICAL)
                                    return props;
                    }
                }
            }
        }
    }
}
```

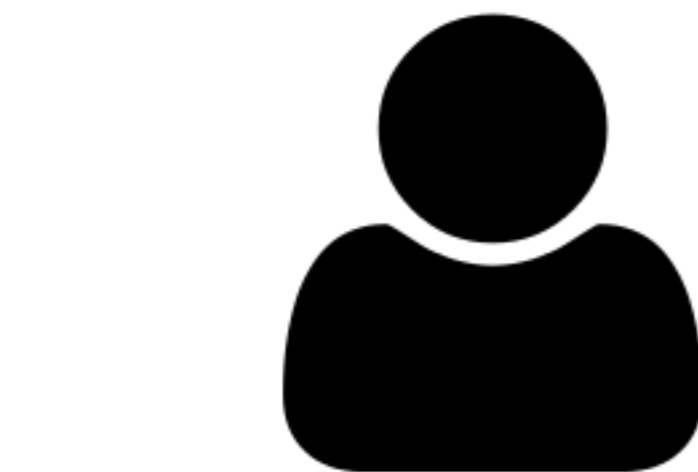
```
symbol.trim());
AlphabetType.CLASSICAL) {
    props);
    components) {
    c.getPreferredSize();
    , d.width);
    , d.height);
    umSize();
    , d.width);
    , d.height);
    c : components) {
    n pref = c.getPreferredSize();
    on min = c.getMinimumSize();

    int width = orient == HORIZONTAL ? pref.width : w;
    int height = orient == VERTICAL ? pref.height : h;

    if (width < min.width)
        width = min.width;
    if (height < min.height)
        height = min.height;

    c.setBounds(x, y, width, height);

    if (orient == HORIZONTAL)
        if (orient == VERTICAL)
            if (orient == HORIZONTAL)
                if (orient == VERTICAL)
                    move=> m s x.
                    rewrite (acc_ssa_state_eq _ (eqxx (get_index x m))).
                    reflexivity.
                    Qed.
```



Project Backup

2017/08/05 **What has been changed in a version?**

Log

```
int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}
for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();
    int x = insets.left;
    int y = insets.top;
```

```
protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();
    genAlphabet(AlphabetType type,
    props) {
    new ArrayList<String>();
    AlphabetType.PROPOSITIONAL) {
    (int) Math.pow(2, props.size());
    j = 0; j < max; j++) {
    g symbol = "";
    = j;
    String prop : props) {
    f (n % 2 == 0)
    symbol += " " + prop;
    lse
```

2017/08/13 **How is a bug in version A fixed in version B?**

Diff

```
c.setBounds(x, y, width, height);
if (orient == HORIZONTAL) {
    if (orient == HORIZONTAL) {
        if (orient == HORIZONTAL) {
            for (Component c : components) {
                Dimension pref = c.getPreferredSize();
                Dimension min = c.getMinimumSize();
                int width = orient == HORIZONTAL ? pref.width : min.width;
                int height = orient == VERTICAL ? pref.height : min.height;
                if (width < min.width)
                    width = min.width;
```

```
symbol.trim());
AlphabetType.CLASSICAL) {
    ops);
    components) {
    c.getPreferredSize();
    , d.width);
    , d.height);
    umSize();
    , d.width);
    , d.height);
    c : components) {
    n pref = c.getPreferredSize();
    on min = c.getMinimumSize();
```

Any support better than file system?

```
if (orient == HORIZONTAL)
}
return props;
}
```

```
c.setBounds(x, y, width, height);
if (orient == HORIZONTAL)
    if (orient == HORIZONTAL)
        if (orient == HORIZONTAL)
            move=> m s x.
            rewrite (acc_ssa_state_eq _ (eqxx (get_index x m))).
            reflexivity.
Qed.
```

Project Backup

2017/08/05 **What has been changed in a version?**

Log

```
int w = 0;
int h = 0;
for (Component c : components) {
    Dimension d = c.getPreferredSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);

    d = c.getMinimumSize();
    w = Math.max(w, d.width);
    h = Math.max(h, d.height);
}
for (Component c : components) {
    Dimension pref = c.getPreferredSize();
    Dimension min = c.getMinimumSize();
    int x = insets.left;
    int y = insets.top;
```

```
if (type == AlphabetType.PROPOSITIONAL) {
    int max = (int) Math.pow(2, props.size());
    for (int j = 0; j < max; j++) {
        String symbol = "";
        int n = j;
```

```
protected Collection<String> genAlphabet(AlphabetType type,
    Collection<String> props) {
    List<String> alphabet = new ArrayList<String>();

    genAlphabet(AlphabetType type,
    props) {
    new ArrayList<String>();
    AlphabetType.PROPOSITIONAL) {
    (int) Math.pow(2, props.size());
    j = 0; j < max; j++) {
    g symbol = "";
    = j;
    String prop : props) {
    f (n % 2 == 0)
    symbol += " " + prop;
    lse
```

2017/08/13 **How is a bug in version A fixed in version B?**

Diff

```
c.setBounds(x, y, w, h);
if (orient == HORIZONTAL) {
    if (orient == HORIZONTAL) {
        if (orient == HORIZONTAL) {
            for (Component c : components)
                Dimension pref = c.getPreferredSize();
                Dimension min = c.getMinimumSize();

            int width = orient == HORIZONTAL ? pref.width : min.width;
            int height = orient == VERTICAL ? pref.height : min.height;

            if (width < min.width)
                width = min.width;
```

```
symbol.trim());
AlphabetType.CLASSICAL) {
    props);

    components) {
    c.getPreferredSize();
    , d.width);
    , d.height);

    umSize();
    , d.width);
    , d.height);

    c : components) {
    n pref = c.getPreferredSize();
    on min = c.getMinimumSize();
```

And how do I manage my code?

Version Control

Version Control

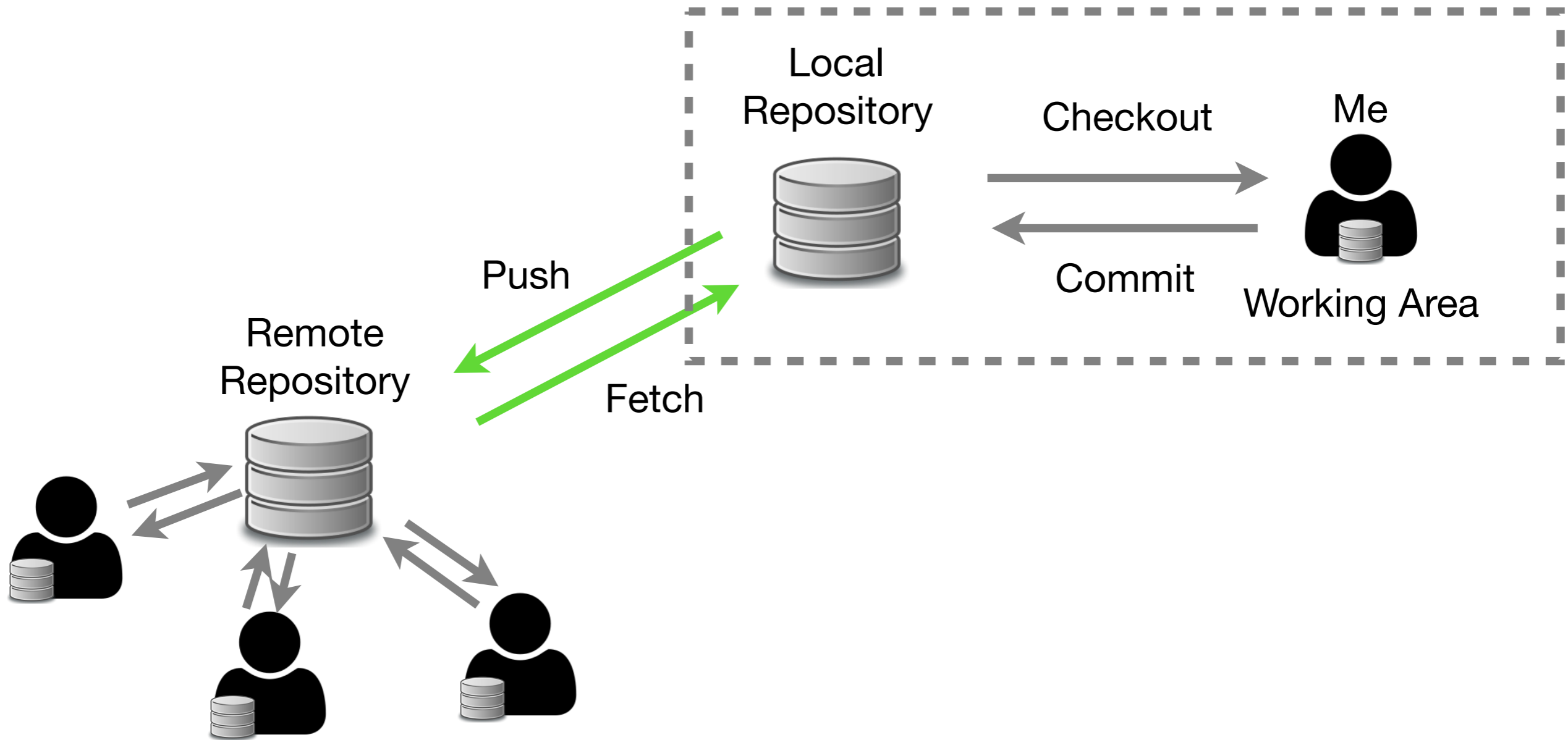
- The developer needs a system that can
 - store files in different versions with log messages,
 - compare files between different versions, and
 - restore files from specific versions.

Version Control

- The developer needs a system that can
 - store files in different versions with log messages,
 - compare files between versions, and
 - restore files from specific versions.

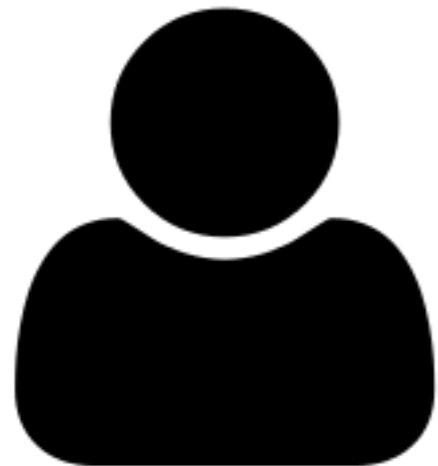


Repository



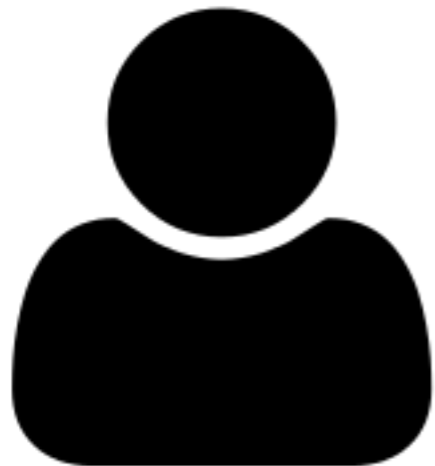
Create Repository

DHCP-22126 project \$ █



Create Repository

```
DHCP-22126 project $ git init
Initialized empty Git repository in /Users/mht208/project/.git/
DHCP-22126 project $ ls
DHCP-22126 project $ ls -a
.  ..  .git
DHCP-22126 project $ █
```



Repository

Create Repository

```
DHCP-22126 project $ git init
Initialized empty Git repository in /Users/mht208/project/.git/
DHCP-22126 project $ ls
DHCP-22126 project $ ls -a
.  ..  .git
DHCP-22126 project $
```

repository

working area (working tree)



Repository

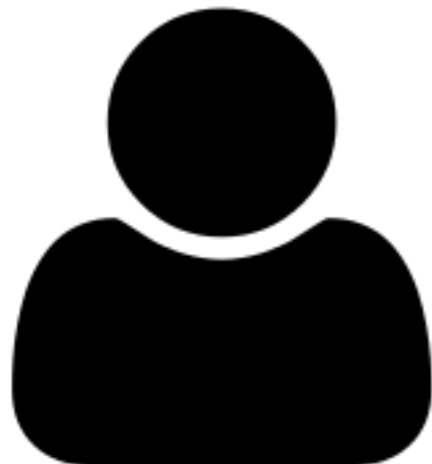
Create Repository

```
DHCP-22126 project $ git init
Initialized empty Git repository in /Users/mht208/project/.git/
DHCP-22126 project $ ls
DHCP-22126 project $ ls -a
.  ..  .git
DHCP-22126 project $
```

repository

working area (working tree)

the default branch is "master"



Repository

File Checkin

```
DHCP-22126 project $ ls -a  
.          ..          .git          README.md  
DHCP-22126 project $ █
```



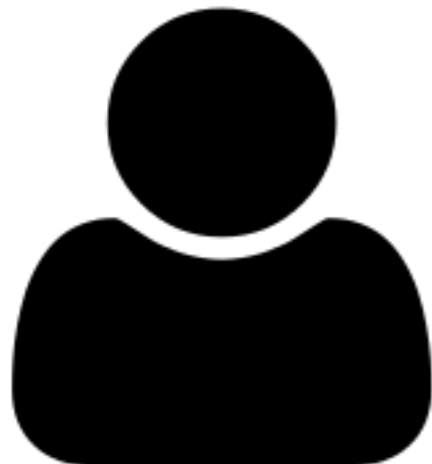
README.md

```
** Convert a DSL state to an SSA state. **  
definition ssa_state (m : vmop) (s : State.k) : bv64SSA.State.t :=  
  fun v =>  
    if (sidx v) == get_index (svar v) m  
    then State.acc (svar v) s  
    else State.acc (svar v) State.empty.  
  ..  
  
mmma ssa_state_equiv :  
  forall m s, state_equiv m s (ssa_state m s).  
  proof.  
    mmma m s x.  
    rewrite (cc.c_ssa_state_eq _ (eqxx (get_index x m))).  
    reflexivity.  
  qed.
```



File Checkin

```
DHCP-22126 project $ ls -a
.          ..          .git      README.md
DHCP-22126 project $ git add README.md
DHCP-22126 project $ git commit
```



README.md

```
** Convert a DSL state to an SSA state. **
definition ssa_state (m : vmop) (s : State.k) : bv64SSA.State.t :=
fun v =>
  if (sidx v) == get_index (svar v) m
  then State.acc (svar v) s
  else State.acc (svar v) State.empty.
  ...

mmma ssa_state_equiv :
  forall m s, state_equiv m s (ssa_state m s).
  proof.
  mmma m s x.
  rewrite (cc.c_ssa_state_eq _ (eqxx (get_index x #))).
  reflexivity.
  id.
```



File Checkin

```
2. git
GNU nano 2.8.5 File: /Users/mht208/project/.git/COMMIT_EDITMSG Modified
Add README.md.
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch
#
# Initial commit
#
# Changes to be committed:
#   new file:   README.md
#
```

[Read 10 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell	^_ Go To Line

DF
.
DF
DF

E.md
.md



```
** Convert a DSL state to an SSA state. **
definition ssa_state (m : vmap) (s : State.k) : bv64SSA.State.t :=
fun v =>
  if (s[ds v] == get_index (svar v) m
  then State.acc (svar v) s
  else State.acc (svar v) State.empty.
  ...

mmma ssa_state_equiv :
  forall m s, state_equiv m s (ssa_state m s).
proof.
  mmma m s x.
  rewrite (cc.c_ssa_state_eq _ (eqxx (get_index x m))).
  reflexivity.
end.
```

README.md



File Checkin

```
2. git
DHCP-22126 project $ git add README.md
DHCP-22126 project $ git commit
[master (root-commit) d05fe5d] Add README.md.
1 file changed, 1 insertion(+)
create mode 100644 README.md
DHCP-22126 project $ █
```

```
[ Read 10 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```



README.md

```
** Convert a DSL state to an SSA state. **
definition ssa_state (m : vmop) (s : State.k) : bv64SSA.State.t :=
fun v =>
if (s[1] v) == get_index (svar v) m
then State.acc (svar v) s
else State.acc (svar v) State.empty.
...
mmma ssa_state_equiv :
forall m s, state_equiv m s (ssa_state m s).
proof.
mmma m s x.
rewrite (acc_ssa_state_eq _ (eqxx (get_index x m))).
reflexivity.
end.
```



File Checkin

```
2. git
DHCP-22126 project $ git add README.md
DHCP-22126 project $ git commit
[master (root-commit) d05fe5d] Add README.md.
1 file changed, 1 insertion(+)
create mode 100644 README.md
DHCP-22126 project $ █
```

```
[ Read 10 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```



```
README.md
** Convert a DSL state to an SSA state. **
definition ssa_state (m : vmop) (s : State.k) : bv64SSA.State.t :=
fun v =>
  if (s[1] v) == get_index (svar v) m
  then State.acc (svar v) s
  else State.acc (svar v) State.empty.
...
mmma ssa_state_equiv :
forall m s, state_equiv m s (ssa_state m s).
proof.
mmma m s x.
rewrite (acc_ssa_state_eq _ (eqxx (get_index x m))).
reflexivity.
end.
```



Add VS Commit

- git add
 - Add files to a staging area
- git commit
 - Add files in the staging area to the repository as a new version



Add VS Commit

- git add
 - Add files to a staging area
- git commit
 - Add files in the staging area to the repository as a new version



If README.md is modified after “git add README.md”, will the modified version be committed after “git commit”?

Add VS Commit

- git add
 - Add files to a staging area
- git commit
 - Add files in the staging area to the repository as a new version



If README.md is modified after “git add README.md”, will the modified version be committed after “git commit”?

NO

History

this is the ID of the version

```
DHCP-22126 project $ git log
commit d05fe5d5bb03bd42abb662e45284020b1c6d85f9
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Sep 12 14:07:39 2017 +0800
```

Add README.md.

```
DHCP-22126 project $ █
```

History

```
4. git

commit 0119516da8f2d96da0695ed9c1a0045616fca9ca
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Jun 27 16:58:41 2017 +0800

    Add bvSubC, bvSbb, and bvSbbC.

commit f5313f6f9174e781e544198fd7ccbb0be4e502f9
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Thu Jun 22 13:31:10 2017 +0800

    add succ_instr and succ_program

commit 886c57392c65723e2741dbe61f79b25e68aa04a1
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Fri Jun 16 08:57:55 2017 +0800

    a script to identify addition chains

commit 0e25d1e7d7ca1c45ce9270c8681056f649865705
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Jun 13 11:54:35 2017 +0800

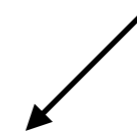
    Fix a bug where ule in Coq becomes ult in OCaml.

commit c9d360dc5609bb9bca06c3dfd0018a59ef59ca08
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Tue Jun 13 11:10:07 2017 +0800

    return exit code from verification

commit 7228396b0dc5e7c71722716cbf52a2b28988bf9e
Merge: c1009c5 372c29a
Author: Bow-Yaw Wang <bywang@iis.sinica.edu.tw>
:
```

is is the ID of the version



```
5284020b1c6d85f9
208@gmail.com>
7 +0800
```

History

```
4. git
commit 0119516da8f2d96da0695ed9c1a0045616fca9ca
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Jun 27 16:58:41 2017 +0800

    Add bvSubC, bvSbb, and bvSbbC.

commit f5313f6f9174e781e544198fd7ccbb0be4e502f9
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Thu Jun 22 13:31:10 2017 +0800

    add succ_instr and succ_program

commit 886c57392c65723e2741dbe61f79b25e68aa04a1
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Fri Jun 16 08:57:55 2017 +0800

    a script to identify addition chains

commit 0e25d1e7d7ca1c45ce9270c8681056f649865705
Author: Ming-Hsien Tsai <mhtsai208@gmail.com>
Date: Tue Jun 13 11:54:35 2017 +0800

    Fix a bug where ule in Coq becomes ult in OCaml.

commit c9d360dc5609bb9bca06c3dfd0018a59ef59ca08
Author: Bow-Yaw <bowyaw@bitbucket.org>
Date: Tue Jun 13 11:10:07 2017 +0800

    return exit code from verification

commit 7228396b0dc5e7c71722716cbf52a2b28988bf9e
Merge: c1009c5 372c29a
Author: Bow-Yaw Wang <bywang@iis.sinica.edu.tw>
:
```

```
4. less
* f6fbbf4 (HEAD -> nondet) Add value type as an argument of store functors.
* 3cdda6c Move conversion between VS and SSAVS to DSL2SSACommons.
* 5be9e28 Add bvDSLRel.
* a479a2a (origin/nondet) Simplify some proofs.
* 4ea01b5 Improve dp_mem and dp_subset.
* a6b4a99 Add SsrFSet to avoid redundant declarations of variable sets.
* 4136533 Remove useless code in bvDSL. Simplify some proofs in bvDSL.
* 73cbe0b Add Equal to Store.
* a73c784 Merge branch 'master' into nondet
|\
| * 1b7952e Add a buggy implementation of fe25519_mul with radix 2^64 representa
tion.
| * 34b2c45 Add simple decision procedures for mem and subset.
| * 0119516 (subtraction) Add bvSubC, bvSbb, and bvSbbC.
* | f5313f6 add succ_instr and succ_program
|/
* 886c573 (origin/ccs17) a script to identify addition chains
* 0e25d1e Fix a bug where ule in Coq becomes ult in OCaml.
* c9d360d return exit code from verification
* 7228396 Merge branch 'master' of fm1.iis.sinica.edu.tw:/home/git/certified_q
hasm_vcg
|\
| * 372c29a Remove unused admitted lemmas in bvSSA2QFBV.
| * 4de5f10 Proved admitted lemmas in bvSSA2zSSA.
| * ca71d71 Add more lemmas.
* | c1009c5 refer to modified gbarith
|/
* c4e55f6 add to_bvdsl.py and rename to_dsl.py
* a5e23c4 allow configurations in comments
* b78c849 Fix a typo.
* 5fd989f Fix a problem in rewrite_zpower_nat.
* 5092d34 Update specifications.
* e7159cd Use Z for equality assertions.
* af5e257 Fix a bug in rewrite_assign2_with.
: 10
```

File Difference

```
DHCP-22126 project $ git diff
diff --git a/README.md b/README.md
index 52cb6cd..eb99592 100644
--- a/README.md
+++ b/README.md
@@ -1,3 @@
+==== About ====
+
This is readme.
DHCP-22126 project $ █
```

File Restore (Checkout)

```
DHCP-22126 project $ cat README.md  
===== About =====
```

This is readme.

```
DHCP-22126 project $ git checkout README.md
```

```
DHCP-22126 project $ cat README.md
```

This is readme.

```
DHCP-22126 project $ █
```



```
protected Collection<String> generatePropositions(AlphabetType type,  
int size) {  
List<String> props = new ArrayList<String>();  
if (type == AlphabetType.POSITIONAL) {  
int base = 25;  
props.add("sym");  
}  
return props;  
}
```

README.md'



File Restore (Checkout)

```
DHCP-22126 project $ cat README.md  
===== About =====
```

This is readme.

```
DHCP-22126 project $ git checkout README.md
```

```
DHCP-22126 project $ cat README.md
```

This is readme.

```
DHCP-22126 project $ █
```



README.md

```
** Convert a DSL state to an SSA state. **  
definition ssa_state (m : vmop) (s : State.k) : bv64SSA.State.t :=  
  fun v =>  
    if (s[id v] == get_index (svar v) m  
    then State.acc (svar v) s  
    else State.acc (svar v) State.empty.  
    ...  
mmma ssa_state_equiv :  
  forall m s, state_equiv m s (ssa_state m s).  
  proof.  
  mmma m s x.  
  rewrite (cc.c_ssa_state_eq _ (eqxx (get_index x #))).  
  reflexivity.  
  id.
```



File Restore (Reset)



README.md”

```
protected Collection<String> genAtomicPropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```

State Area

```
protected Collection<String> genAtomicPropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```



README.md

```
** Convert a DSL state to an SSA state. *)
definition ssa_state (m : vmop) (s : State.t) : bv64SSA.State.t :=
fun v =>
if (sIdx v) == get_index (svar v) m
then State.occ (svar v) s
else State.occ (svar v) State.empty.
...
lemma ssa_state_equiv :
forall! m s, state_equiv m s (ssa_state m s).
proof.
move=> m s x.
rewrite Gcc.ssa_state_eq _ (eqx (get_index x m)).
reflexivity.
qed.
```

File Restore (Reset)

\$ git checkout README.md



README.md”

```
protected Collection<String> genAtomicPropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```




File Restore (Reset)

\$ git checkout README.md



README.md'

```
protected Collection<String> genAtomicPropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```



```
protected Collection<String> genAtomicPropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```



README.md

```
** Convert a DSL state to an SSA state. *)
definition ssa_state (m : vmop) (s : State.t) : bv64SSA.State.t :=
fun v =>
if (sidx v) == get_index (svar v) m
then State.occ (svar v) s
else State.occ (svar v) State.empty.
...
lemma ssa_state_equiv :
forall! m s, state_equiv m s (ssa_state m s).
proof.
move!> m s k.
rewrite! Gcc.ssa_state_eq _ (eqx (get_index x m)).
reflexivity.
qed.
```

File Restore (Reset)



README.md”

```
protected Collection<String> genAtomicPropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```

State Area

```
protected Collection<String> genAtomicPropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```



README.md

```
** Convert a DSL state to an SSA state. *)
definition ssa_state (m : vmop) (s : State.t) : bv64SSA.State.t :=
fun v =>
if (sIdx v) == get_index (svar v) m
then State.occ (svar v) s
else State.occ (svar v) State.empty.
lemma ssa_state_equiv :
forall! m s, state_equiv m s (ssa_state m s).
proof.
move=> m s x.
rewrite Gcc.ssa_state_eq _ (eqx (get_index x m)).
reflexivity.
qed.
```

File Restore (Reset)

\$ git reset README.md



README.md"

```
protected Collection<String> generatePropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```

State (G) area (X)

```
protected Collection<String> generatePropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```



README.md

```
** Convert a DSL state to an SSA state. *)
definition ssa_state (m : vmop) (s : State.t) : bv64SSA.State.t :=
fun v =>
if (sidx v) == get_index (svar v) m
then State.occ (svar v) s
else State.occ (svar v) State.empty.
...
lemma ssa_state_equiv :
forall! m s, state_equiv m s (ssa_state m s).
proof.
move=> m s X.
rewrite Gcc.ssa_state_eq _ (eqx (get_index x m)).
reflexivity.
qed.
```

File Restore (Reset)

```
$ git reset README.md
```



README.md”

```
protected Collection<String> generatePropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```

Staging Area
(Git Index)



File Restore (Reset)

```
$ git reset README.md
```

```
$ git checkout README.md
```



README.md”

```
protected Collection<String> generatePropositions(AlphabetType type,
int size) {
List<String> props = new ArrayList<String>();
if (type == AlphabetType.PROPOSITIONAL) {
int base = 15;
props.add(sym);
}
return props;
}
```

Staging Area
(Git Index)



File Restore (Reset)

```
$ git reset README.md
```

```
$ git checkout README.md
```



README.md

```
** Convert a DSL state to an SSA state. *)
definition ssa_state (m : vmop) (s : State.t) : bv64SSA.State.t :=
fun v =>
  if (sidx v) == get_index (svar v) m
  then State.occ (svar v) s
  else State.occ (svar v) State.empty.

lemma ssa_state_equiv :
forall! m s, state_equiv m s (ssa_state m s).
proof.
move! m s k.
rewrite (occ_ssa_state_eq _ (eqx (get_index x #0))).
reflexivity.
end.
```

**Staging Area
(Git Index)**

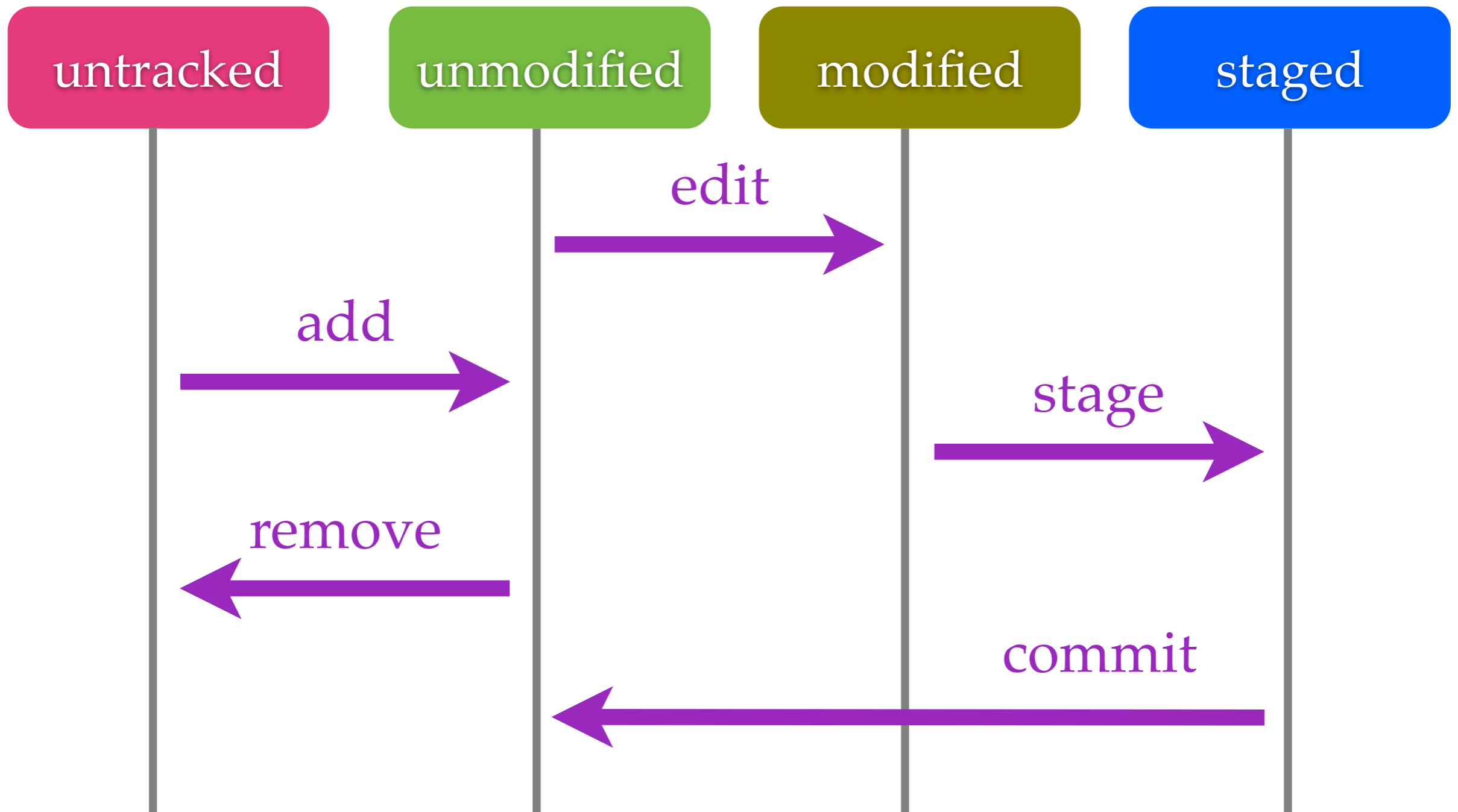


README.md

```
** Convert a DSL state to an SSA state. *)
definition ssa_state (m : vmop) (s : State.t) : bv64SSA.State.t :=
fun v =>
  if (sidx v) == get_index (svar v) m
  then State.occ (svar v) s
  else State.occ (svar v) State.empty.

lemma ssa_state_equiv :
forall! m s, state_equiv m s (ssa_state m s).
proof.
move! m s k.
rewrite (occ_ssa_state_eq _ (eqx (get_index x #0))).
reflexivity.
end.
```

File Status



File Status

```
DHCP-22126 project1 $ git status
```

```
On branch master
```

```
Your branch is ahead of 'origin/master' by 3 commits.
```

```
(use "git push" to publish your local commits)
```

```
Changes to be committed:
```

```
(use "git reset HEAD <file>..." to unstage)
```

```
    modified:   B.c
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   A.c
```

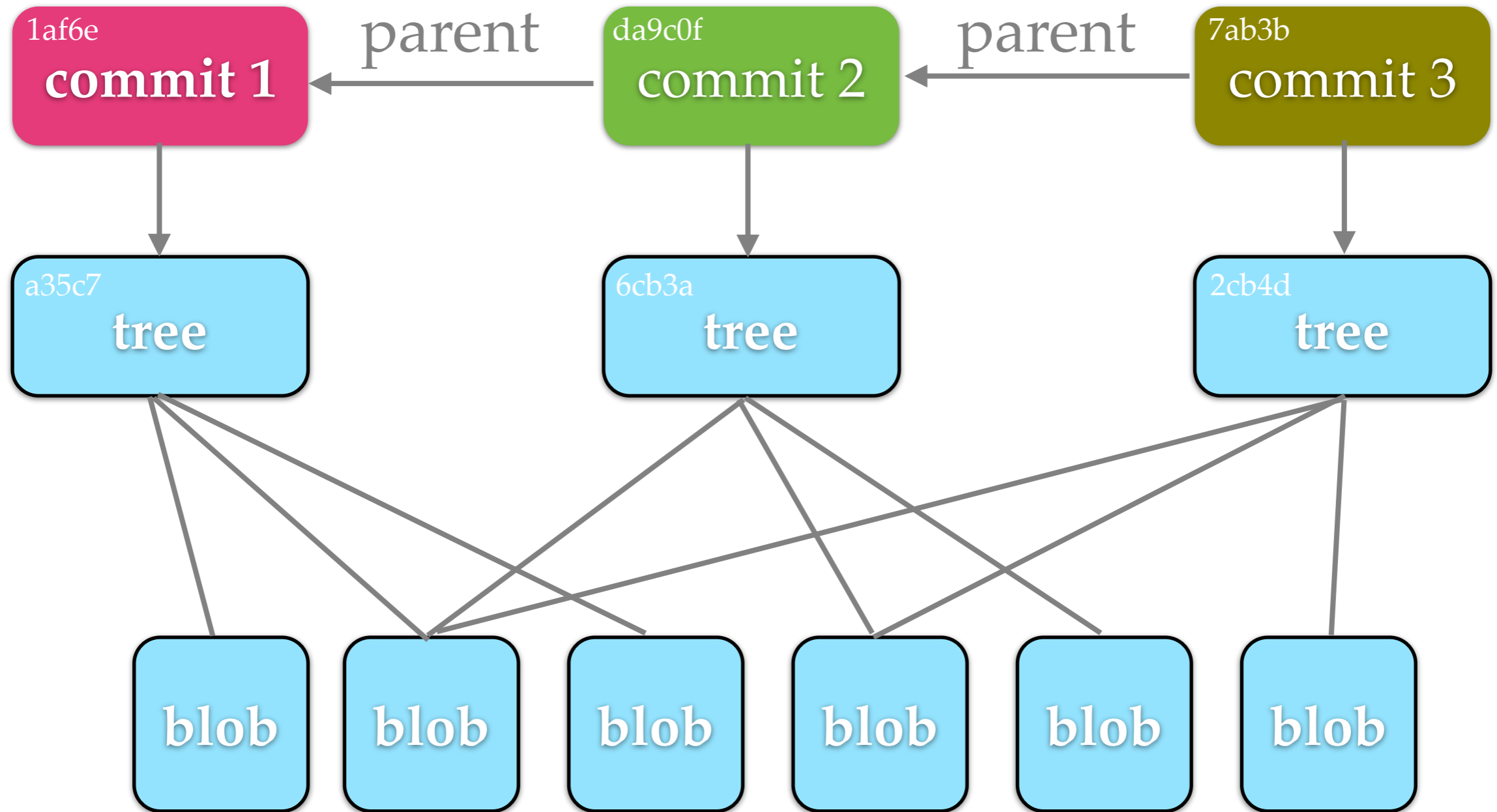
```
    modified:   README.md
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
    D.java
```

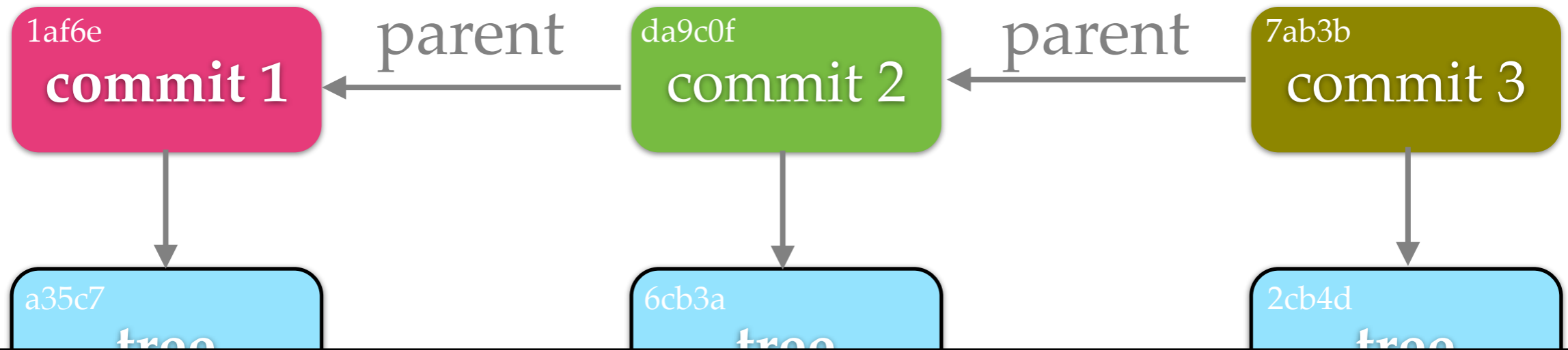
Data Model



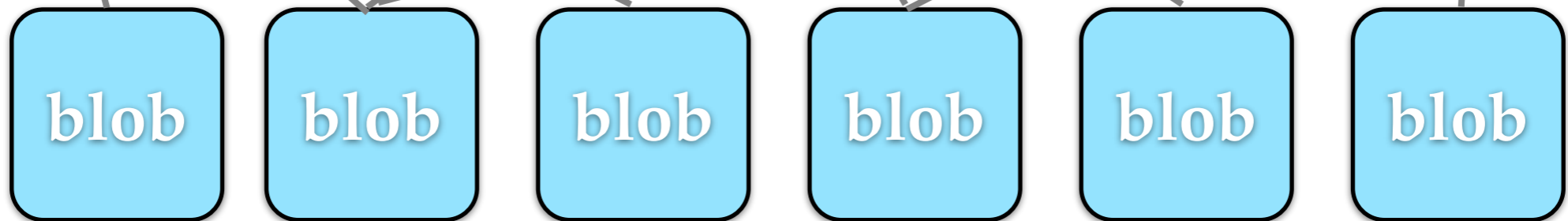
Data compressed by ZLIB

SHA1 hash of data as file name

Data Model



A commit is a pointer (with additional data)



Data compressed by ZLIB

SHA1 hash of data as file name

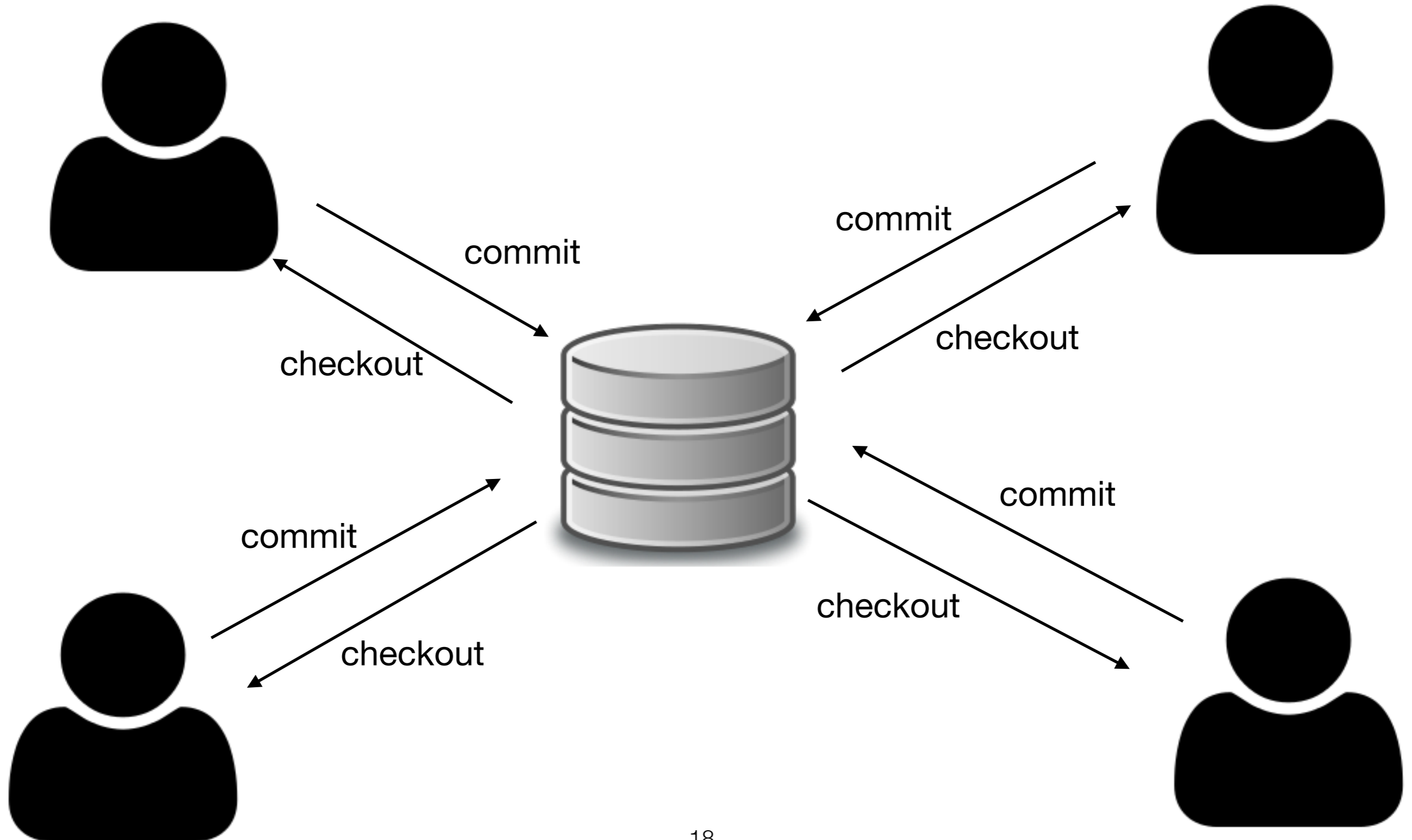
Version Pointers

- Commits
- Branches
- Tags
- HEAD

Local Version Control

- SCCS (Source Code Control System, 1972)
- RCS (Revision Control System, 1982)

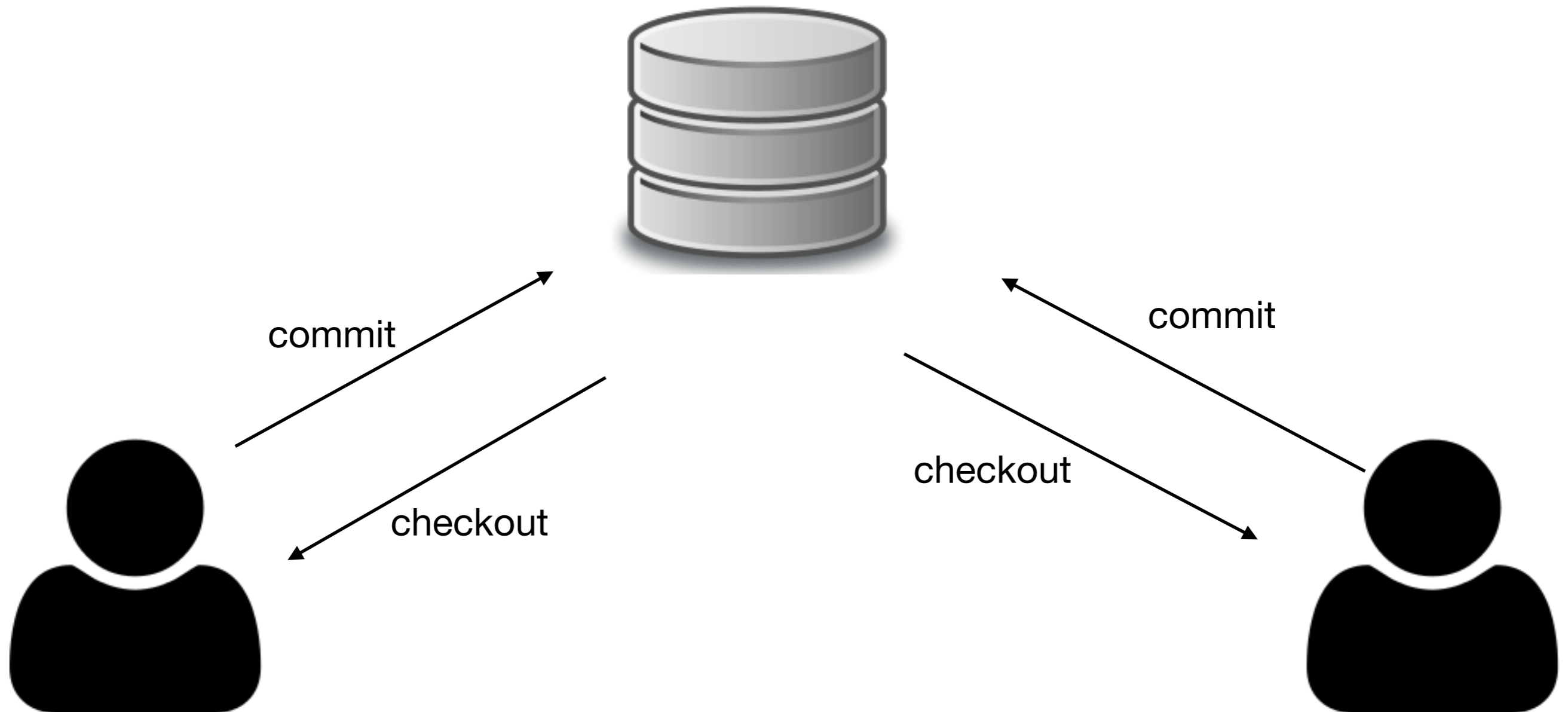
Multiple Developers



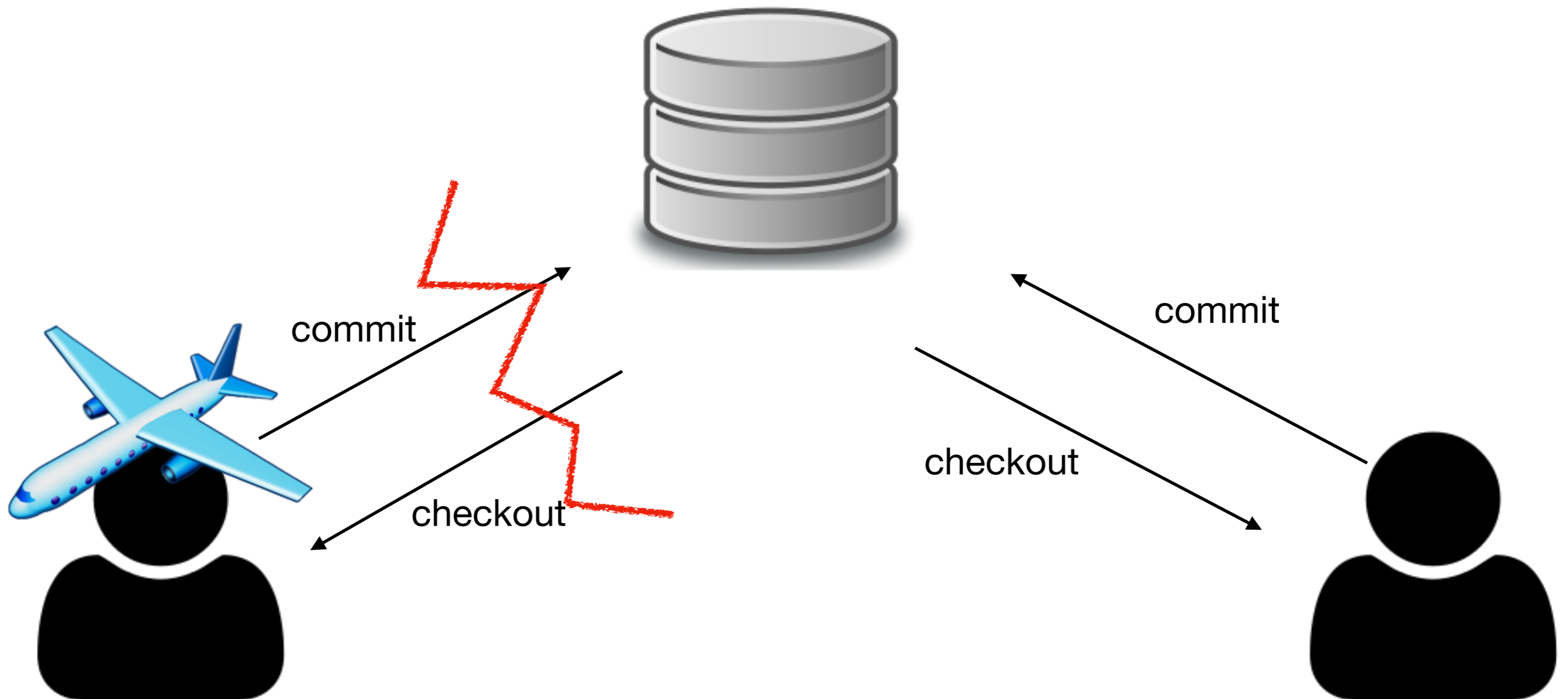
Centralized Version Control

- CVS (Concurrent Versions System, 1990)
- SVN (Subversion, 2000)
- ClearCase
- Visual SourceSafe

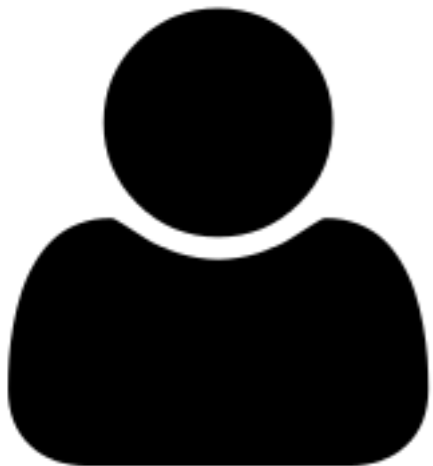
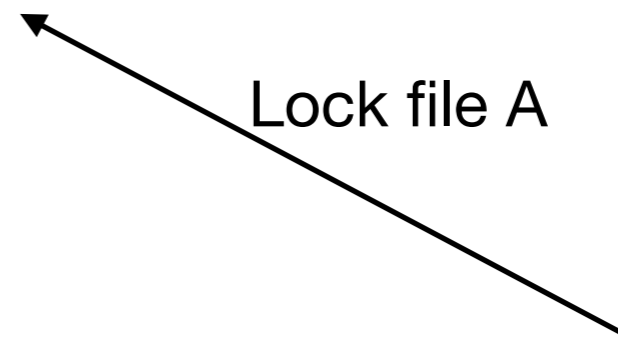
Problems



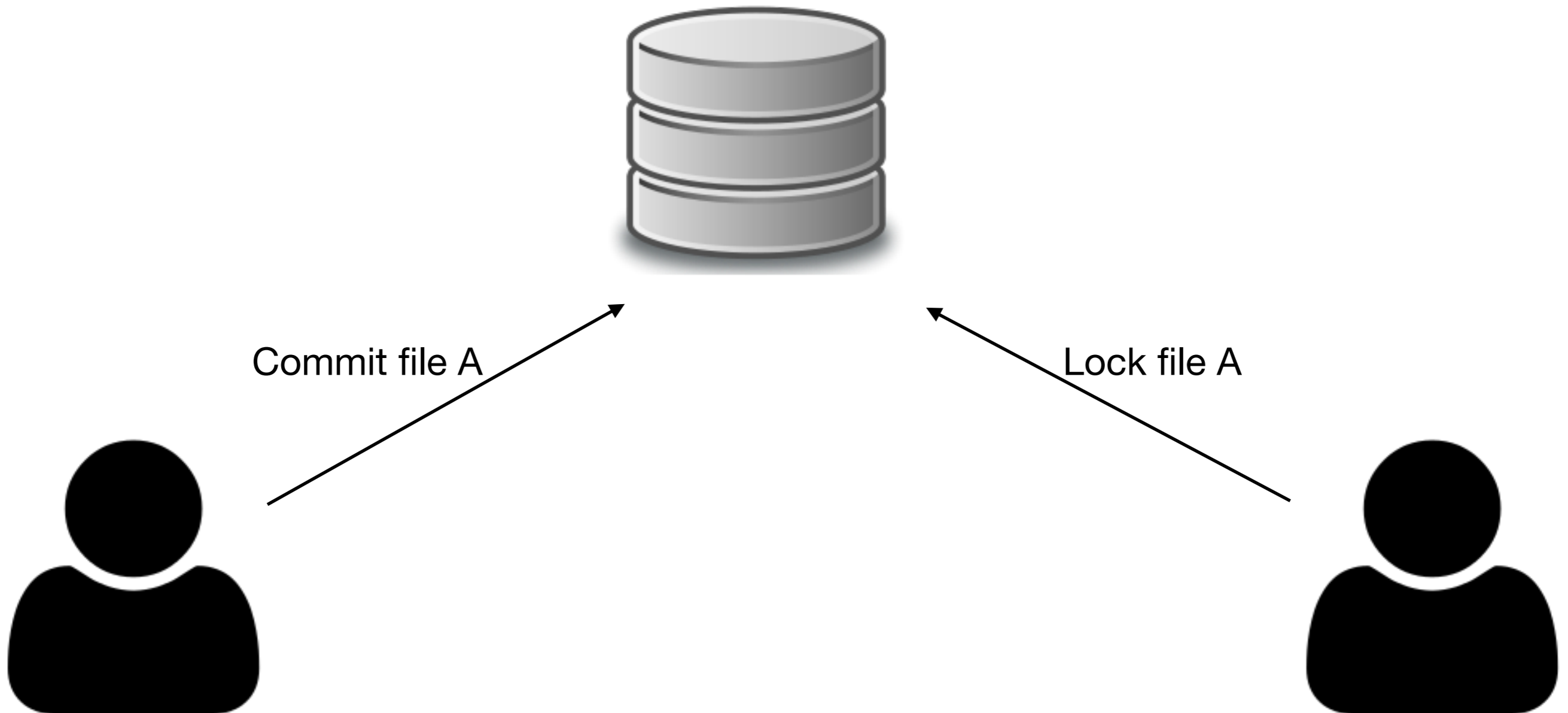
Problems



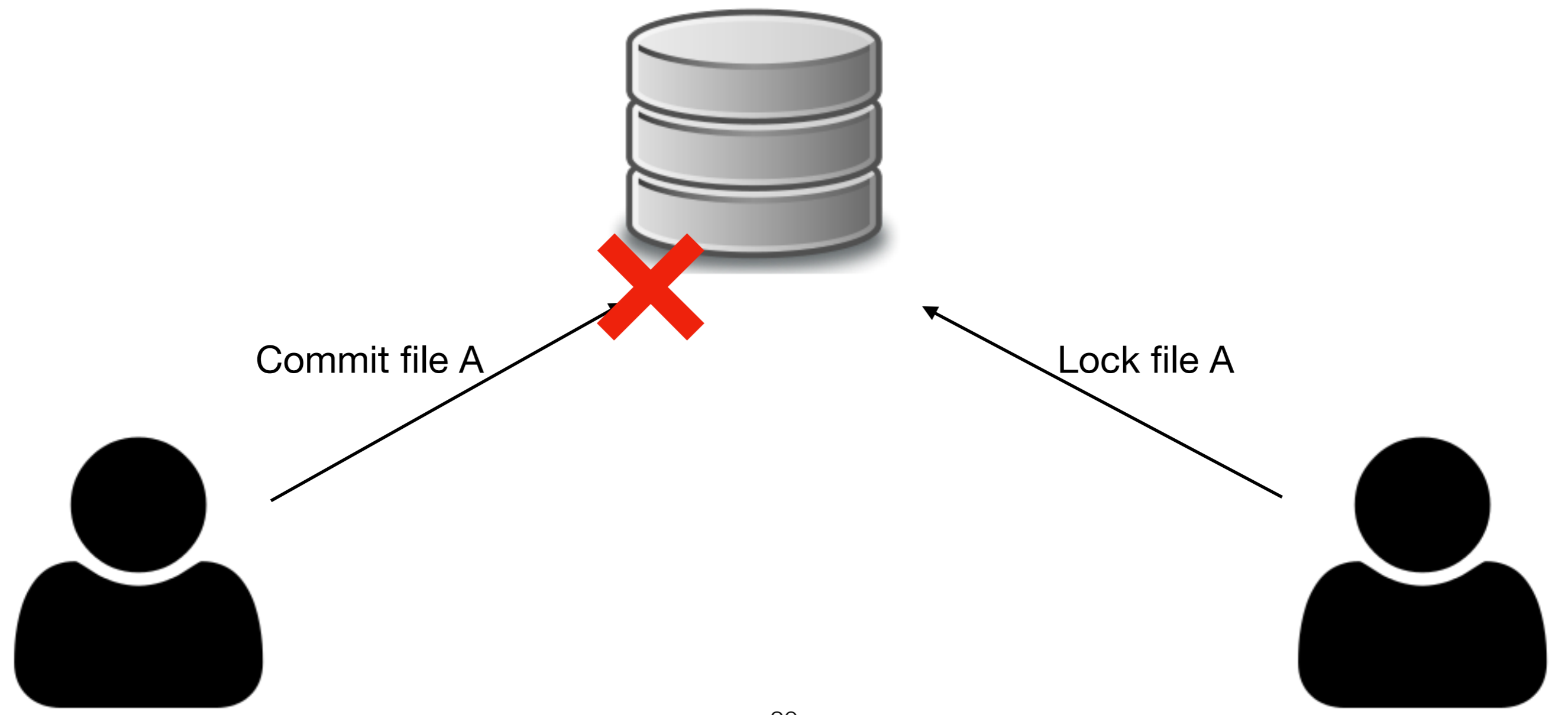
Problems



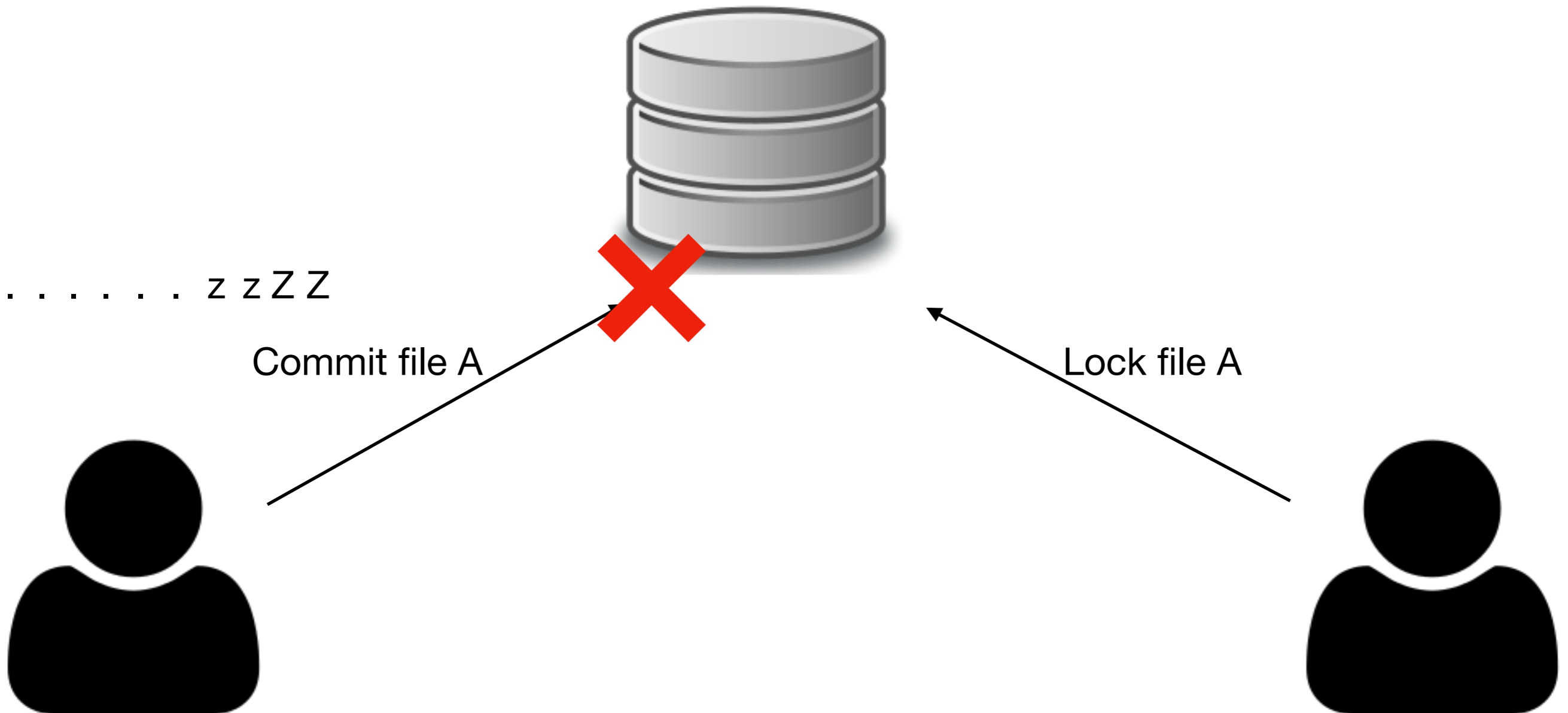
Problems



Problems



Problems



Distributed Approach



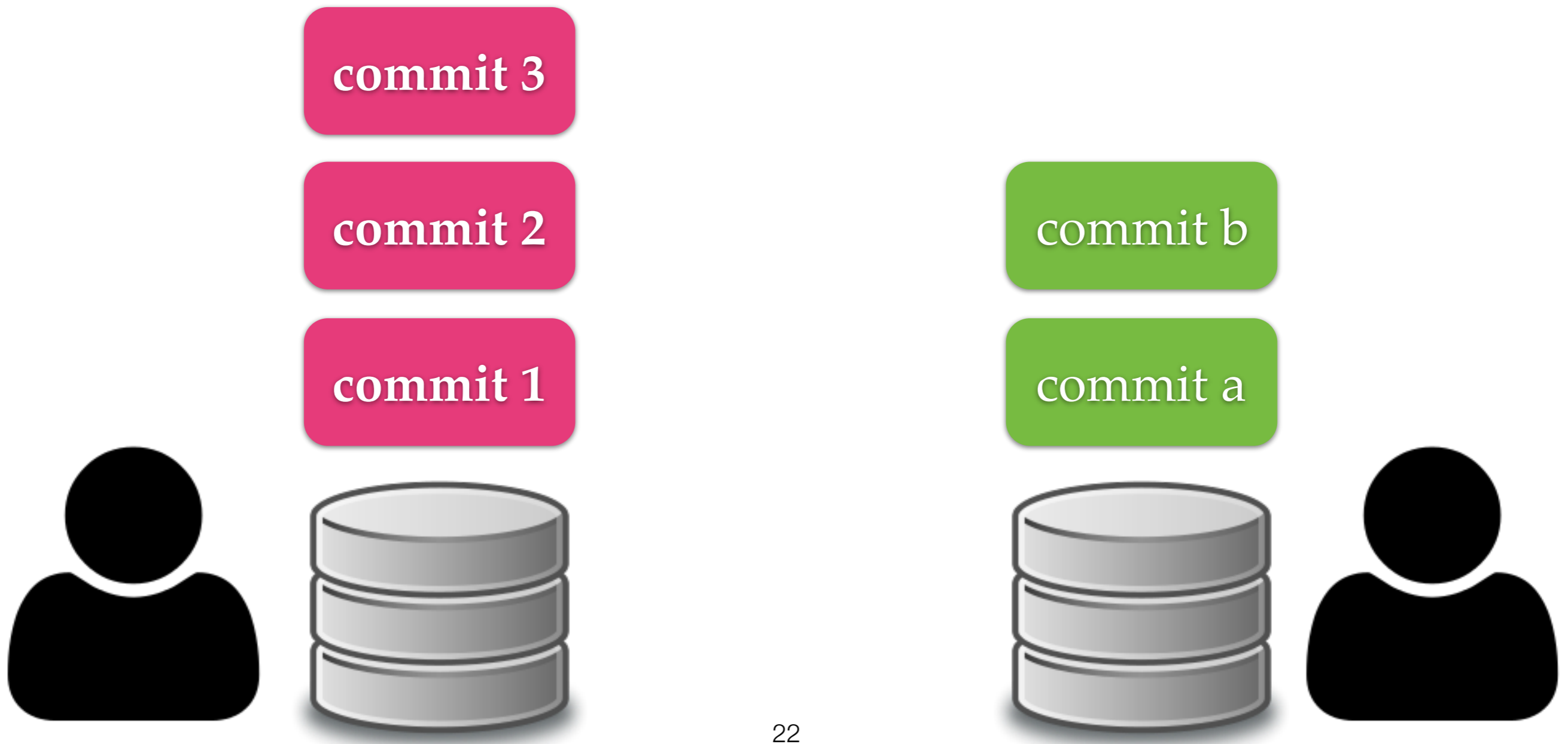
Distributed Approach

Supported protocols: ssh, git, http, and https

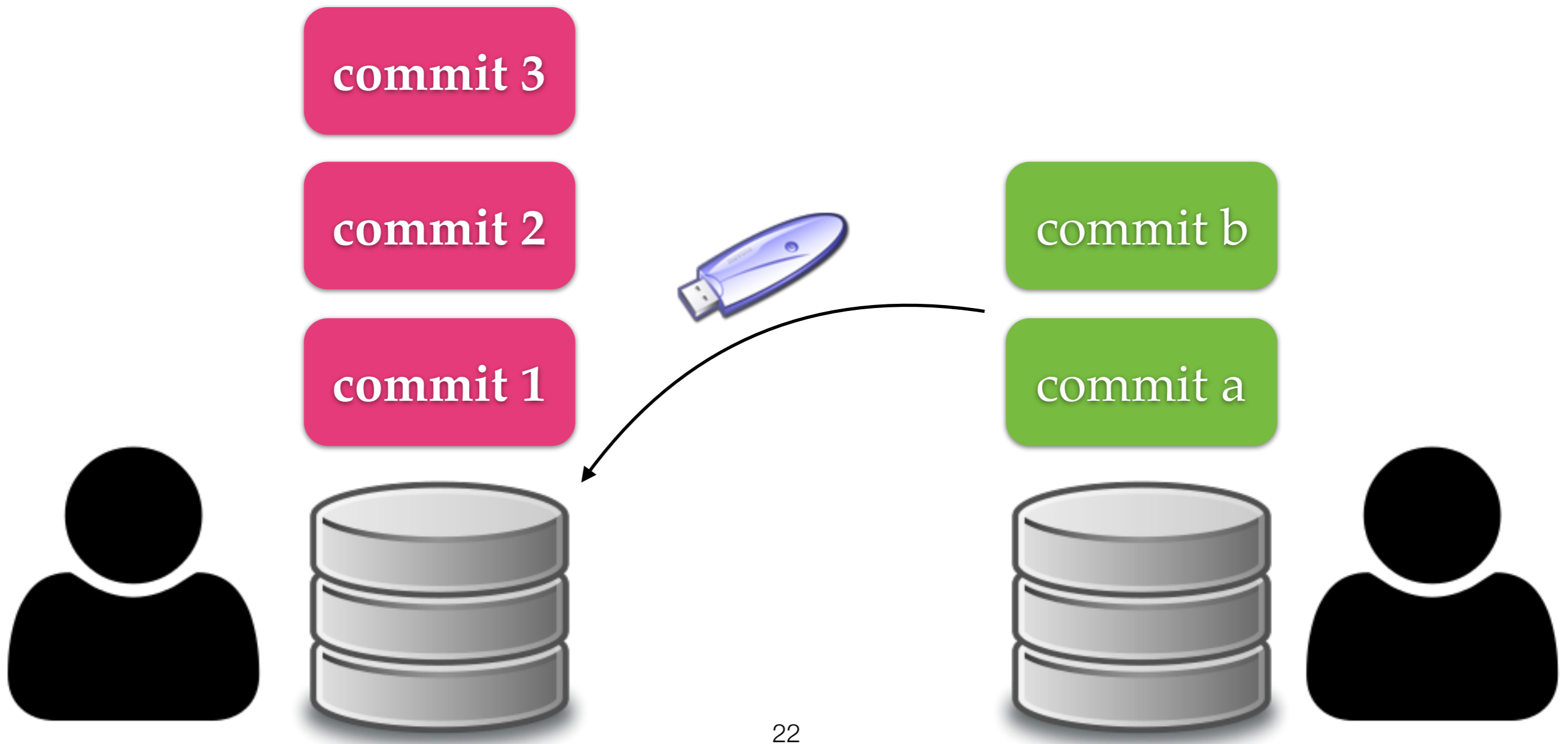
```
DHCP-22126 demo $ ls
DHCP-22126 demo $ git clone https://github.com/atom/atom.git
Cloning into 'atom'...
remote: Counting objects: 179291, done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 179291 (delta 13), reused 13 (delta 10), pack-reused 179265
Receiving objects: 100% (179291/179291), 285.63 MiB | 4.65 MiB/s, done.
Resolving deltas: 100% (127760/127760), done.
DHCP-22126 demo $ ls
atom
DHCP-22126 demo $ █
```



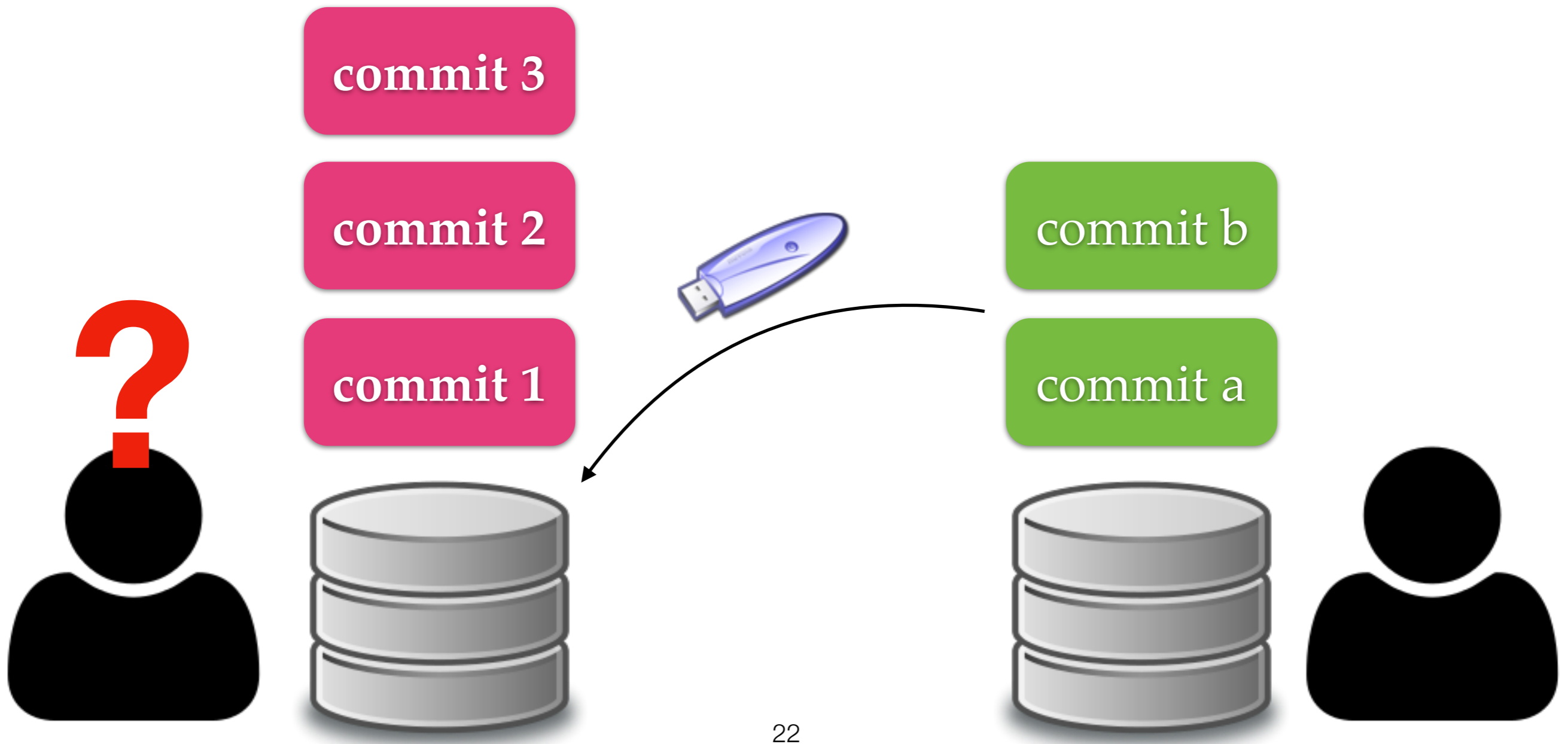
Synchronization



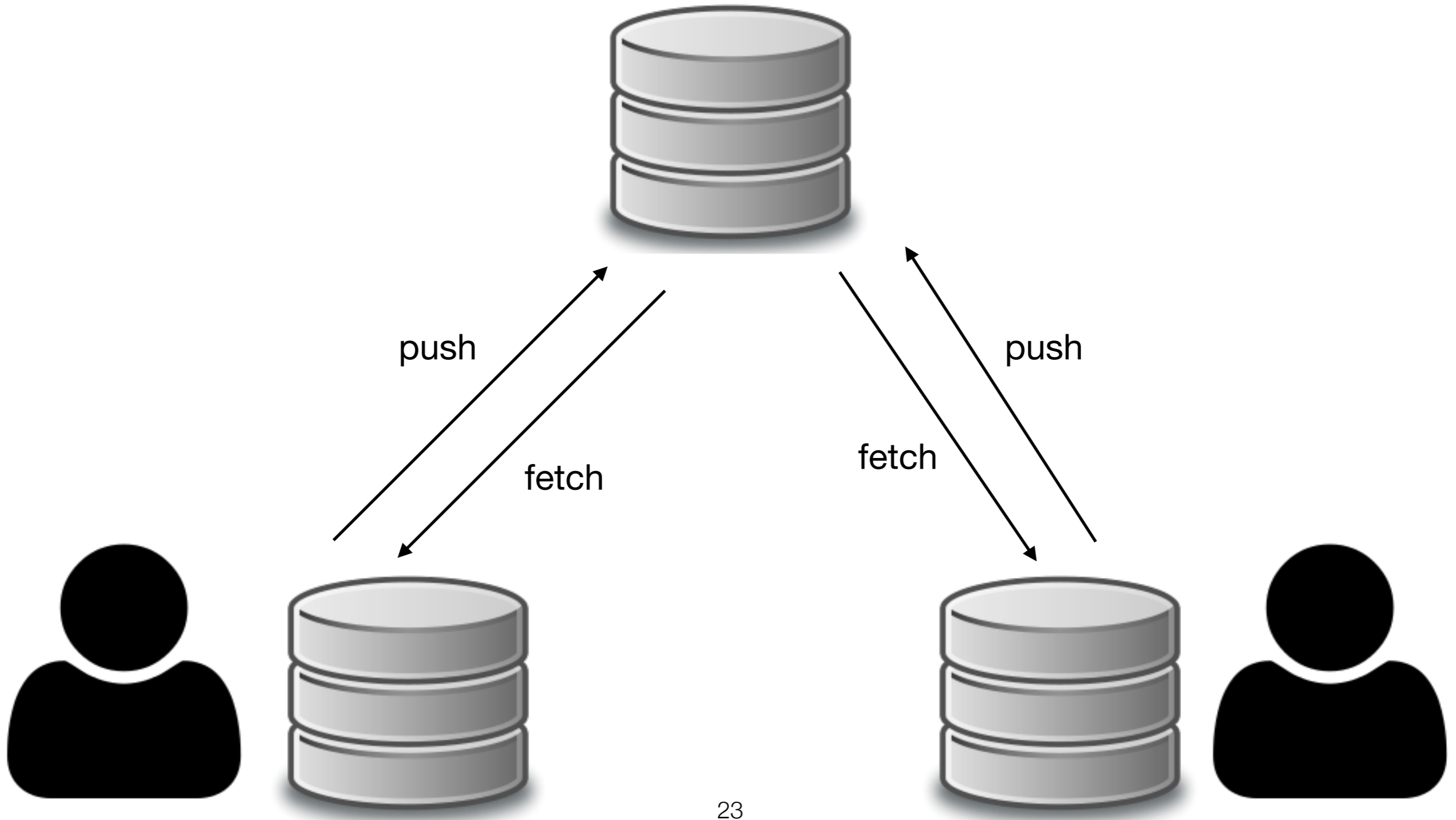
Synchronization



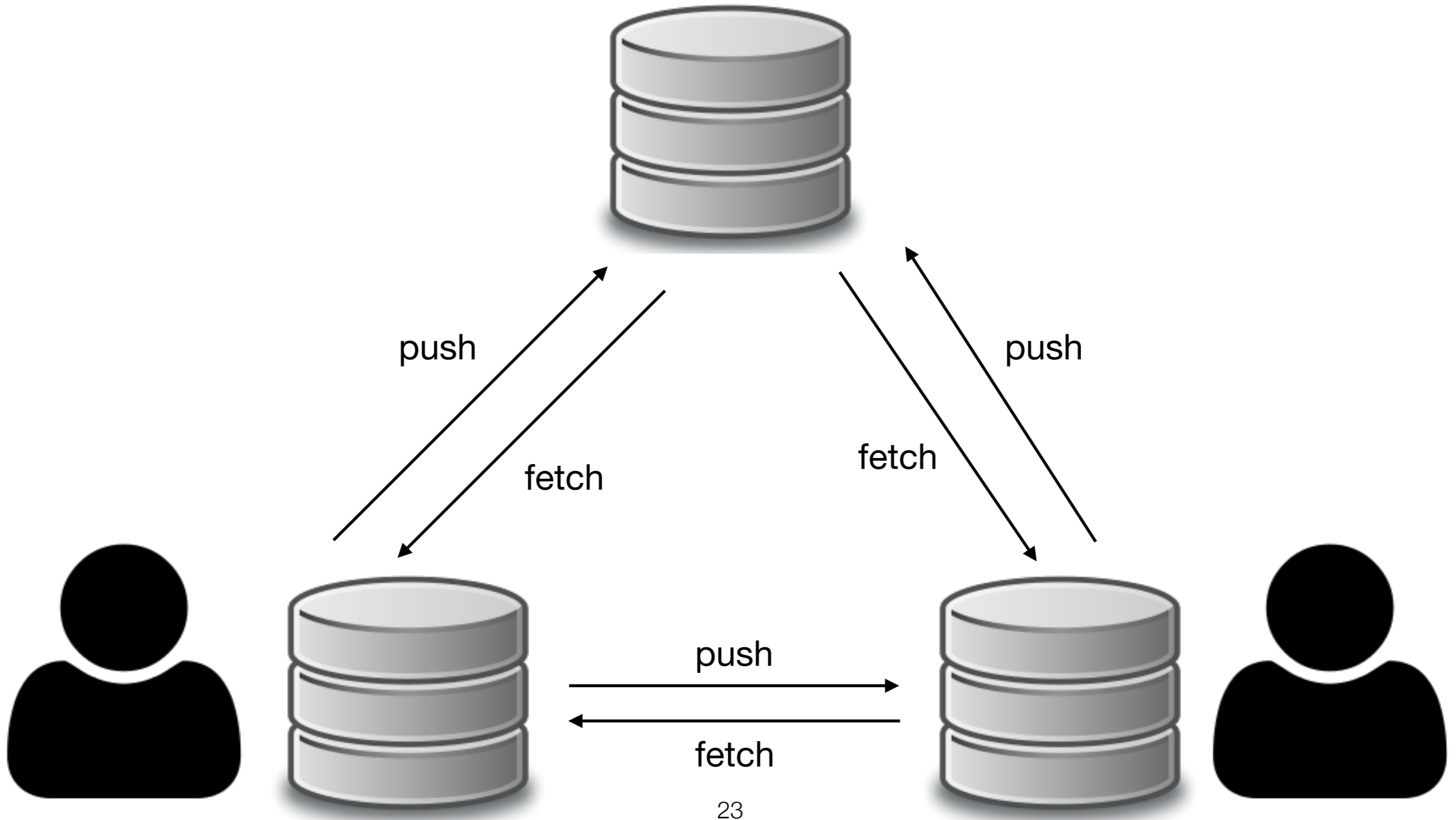
Synchronization



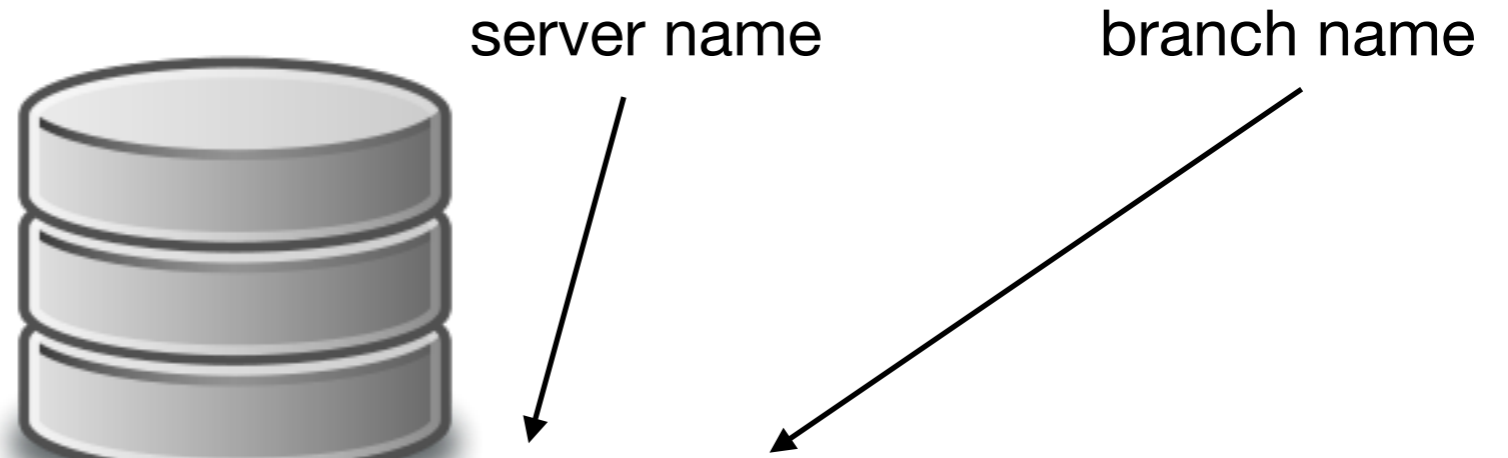
Fetch and Push



Fetch and Push



Fetch and Push



```
DHCP-22126 project2 $ git push origin master
```

```
Counting objects: 3, done.
```

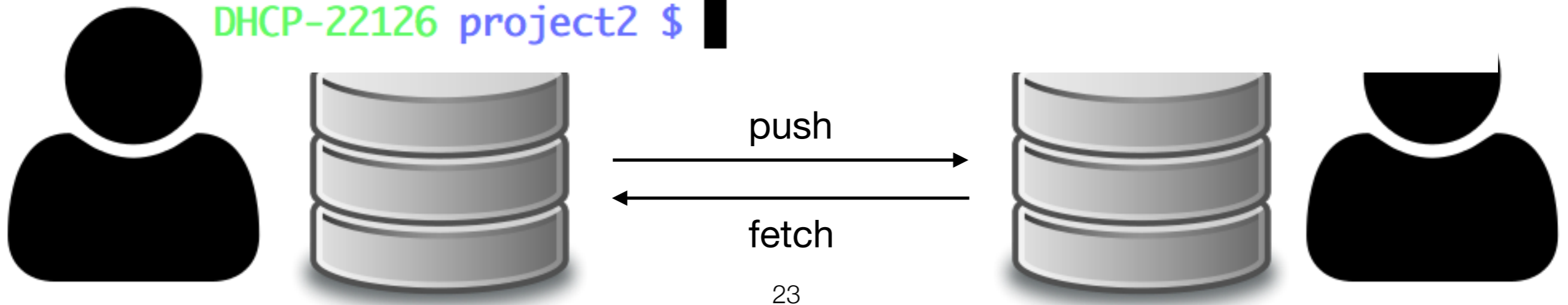
```
Writing objects: 100% (3/3), 288 bytes | 0 bytes/s, done.
```

```
Total 3 (delta 0), reused 0 (delta 0)
```

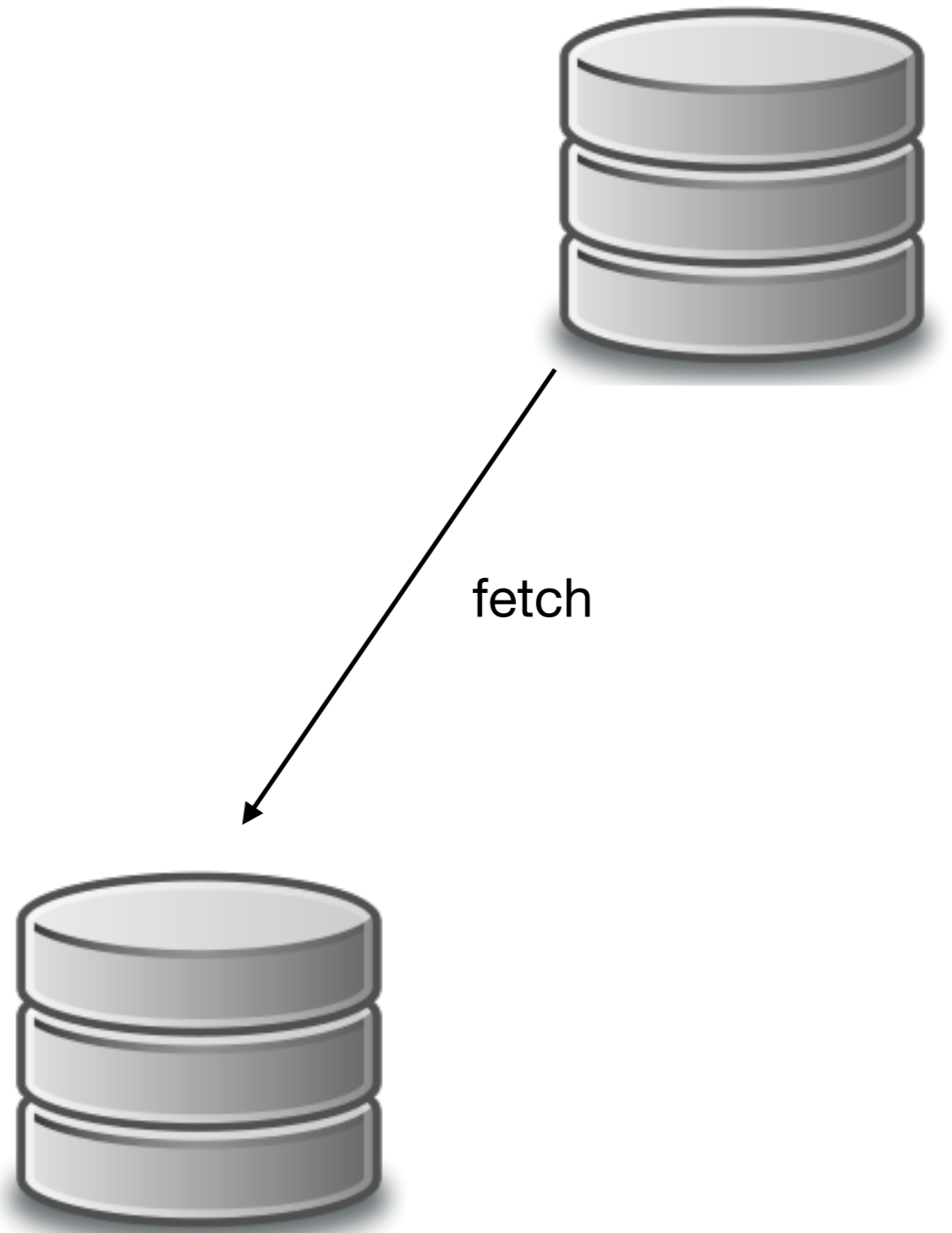
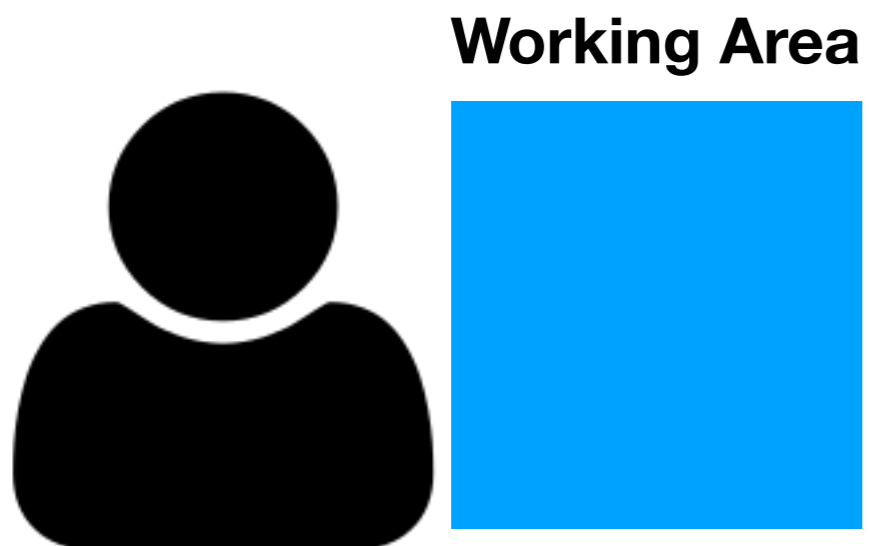
```
To /Users/mht208/demo/server
```

```
3691b0d..57d4f6e master -> master
```

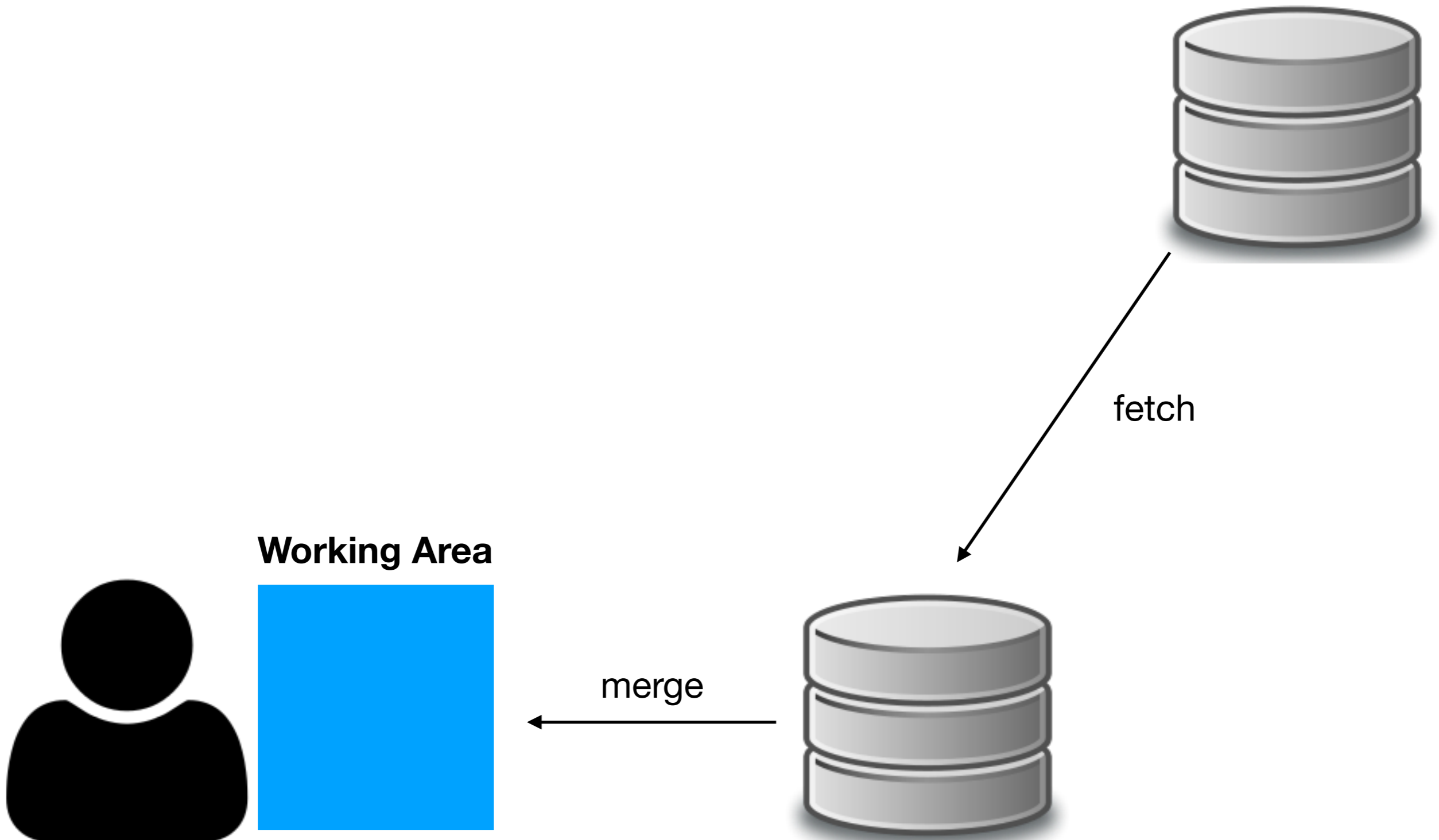
```
DHCP-22126 project2 $ █
```



Merge

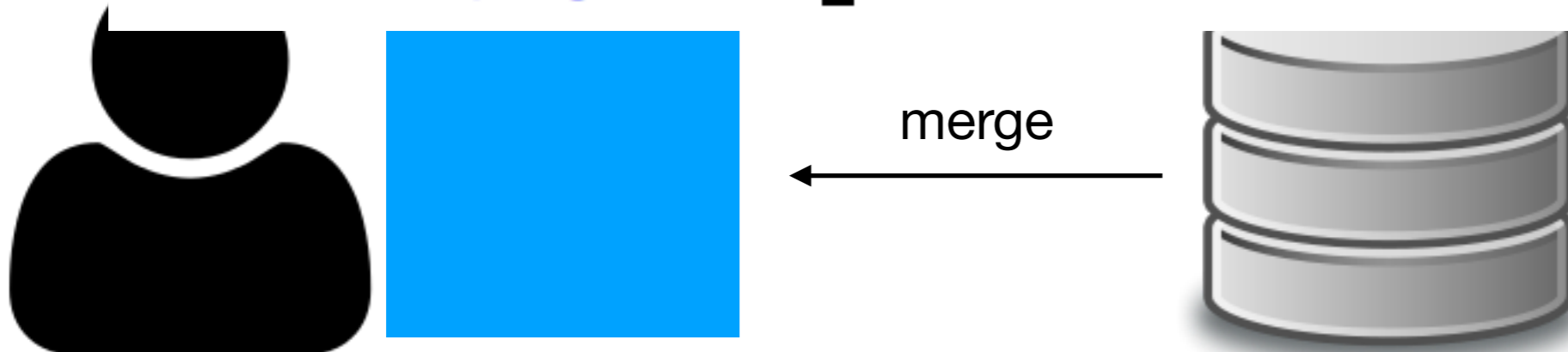


Merge



Merge

```
DHCP-22126 project1 $ git fetch
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From /Users/mht208/demo/server
    3691b0d..57d4f6e  master    -> origin/master
DHCP-22126 project1 $ git merge
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
DHCP-22126 project1 $ █
```

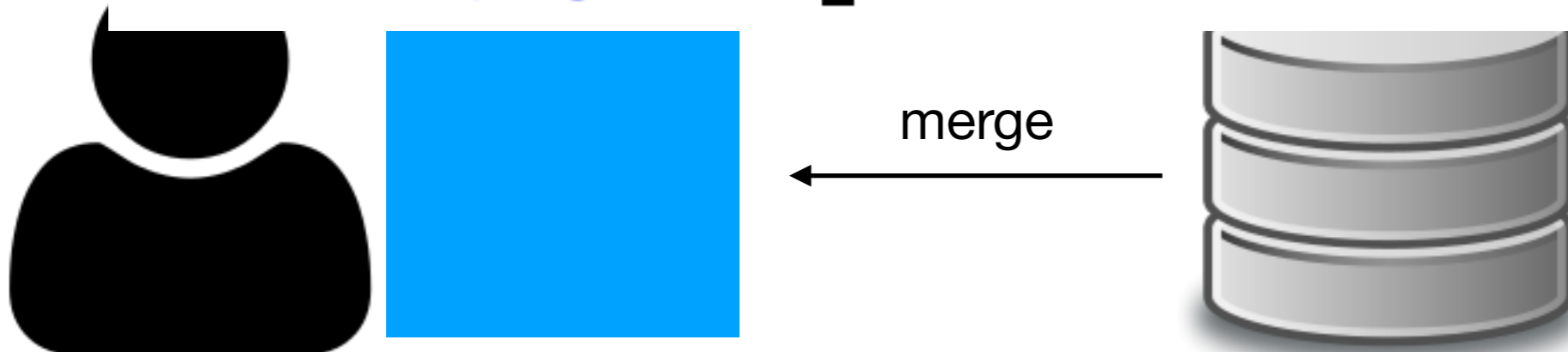


Merge

```
DHCP-22126 project1 $ git fetch
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From /Users/mht208/demo/server
```

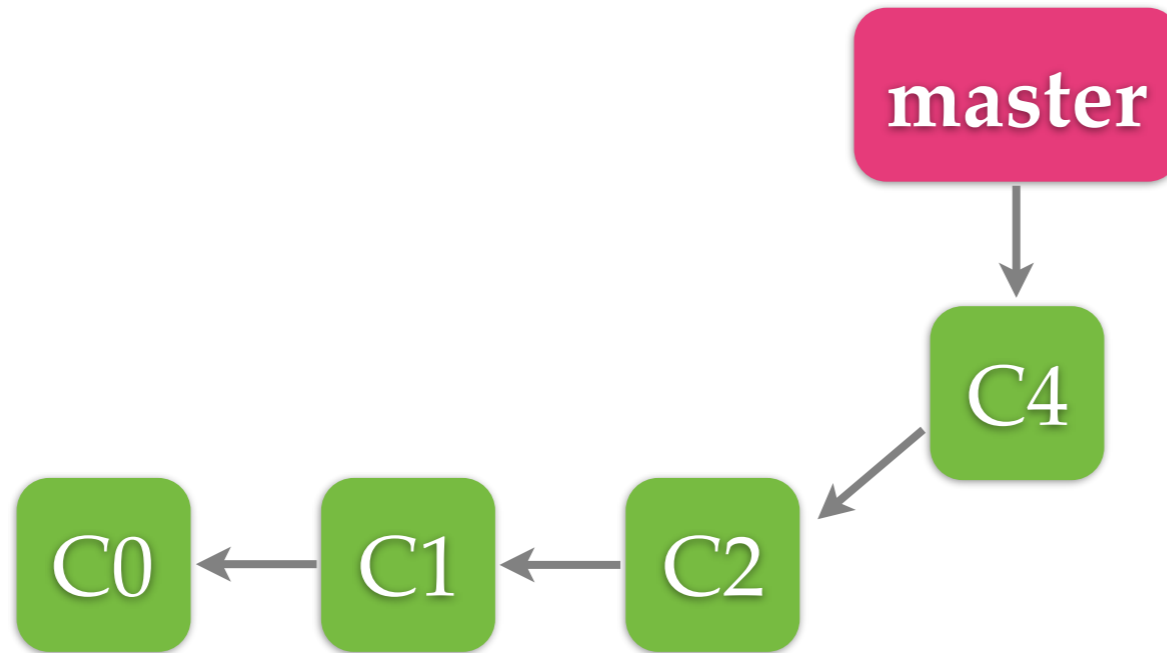
```
36 git pull = git fetch + git merge
DHCP-22126 project1 $ git merge
```

```
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
DHCP-22126 project1 $ █
```

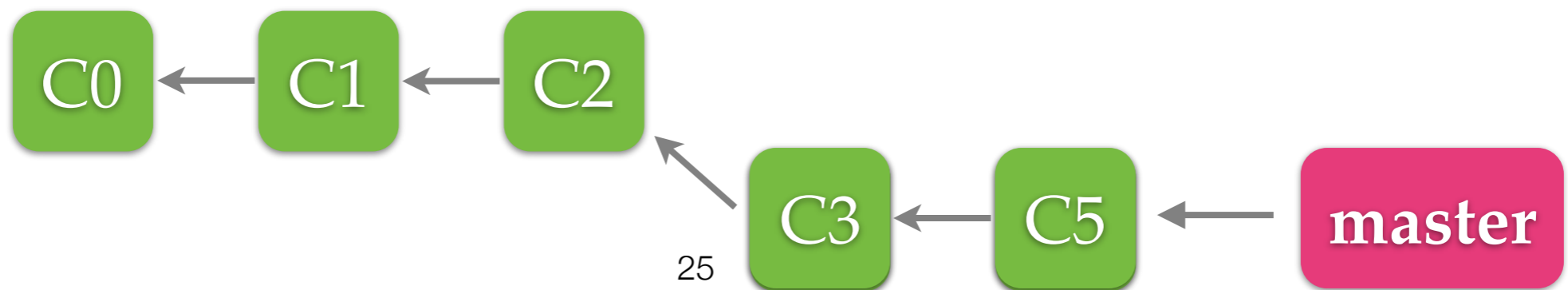


Merge

Working Area

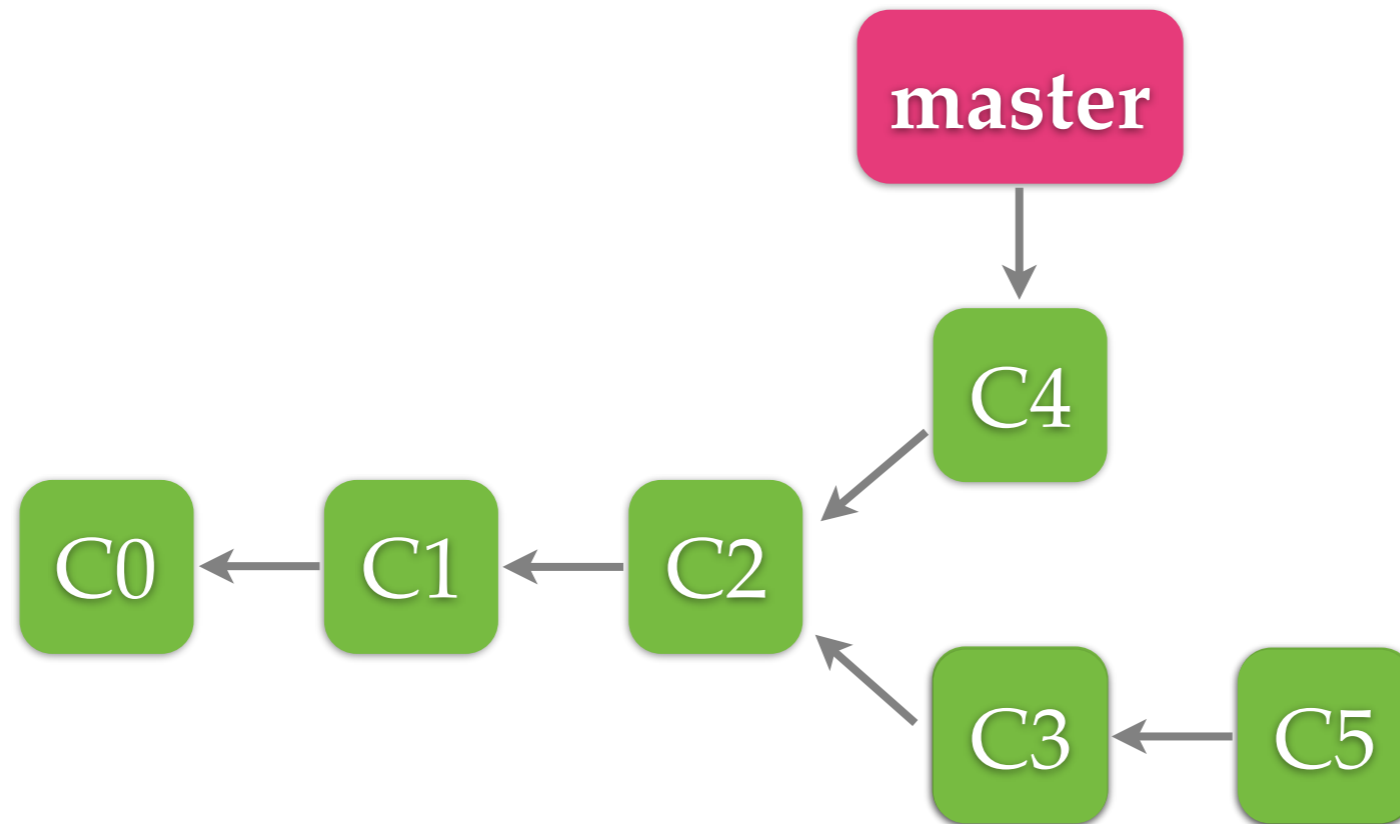


Repository

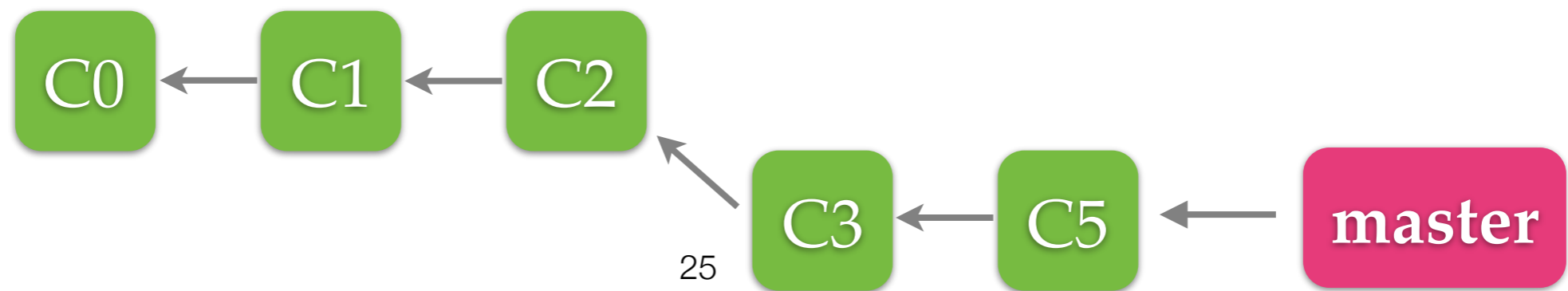


Merge

Working Area

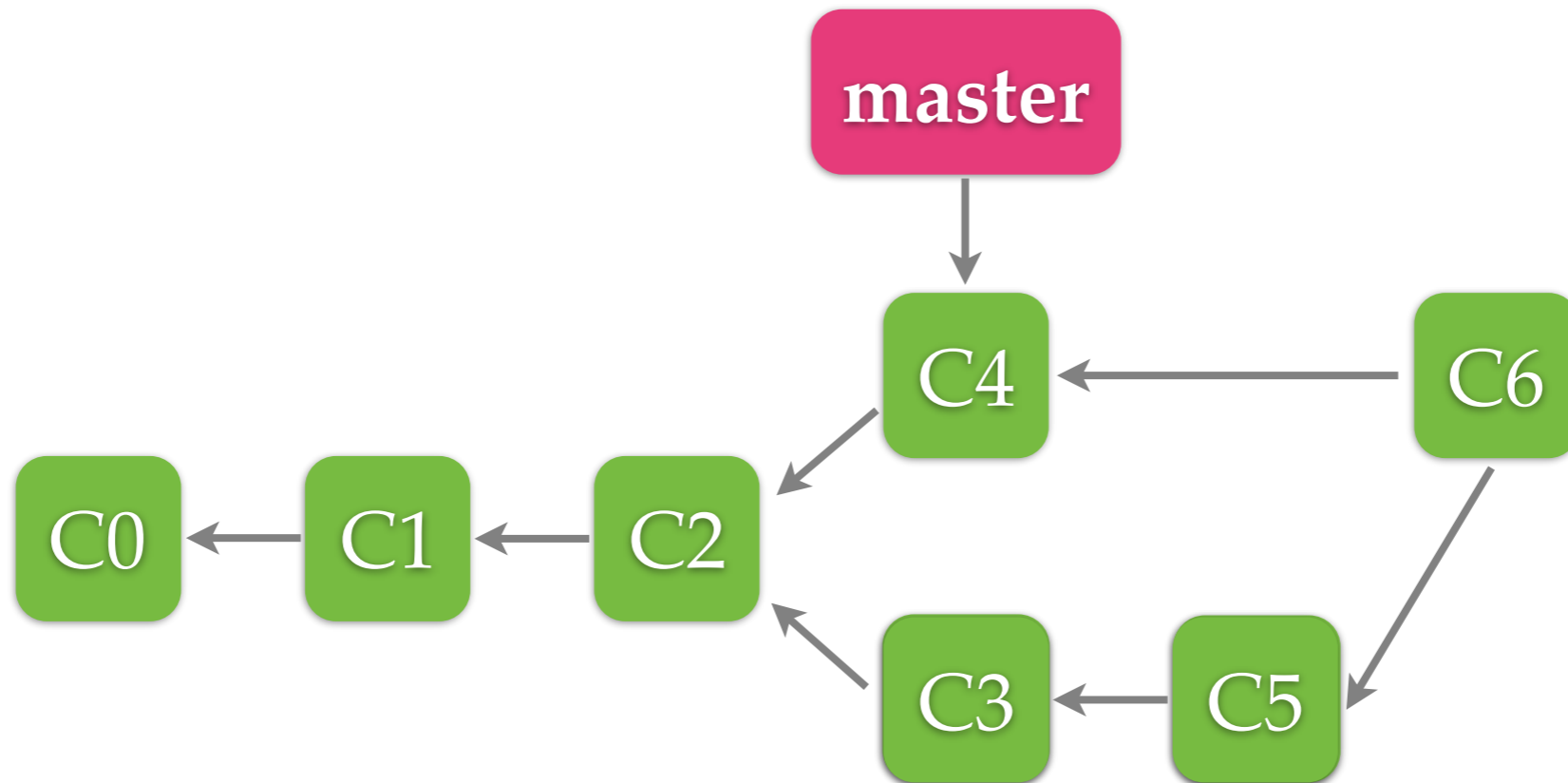


Repository

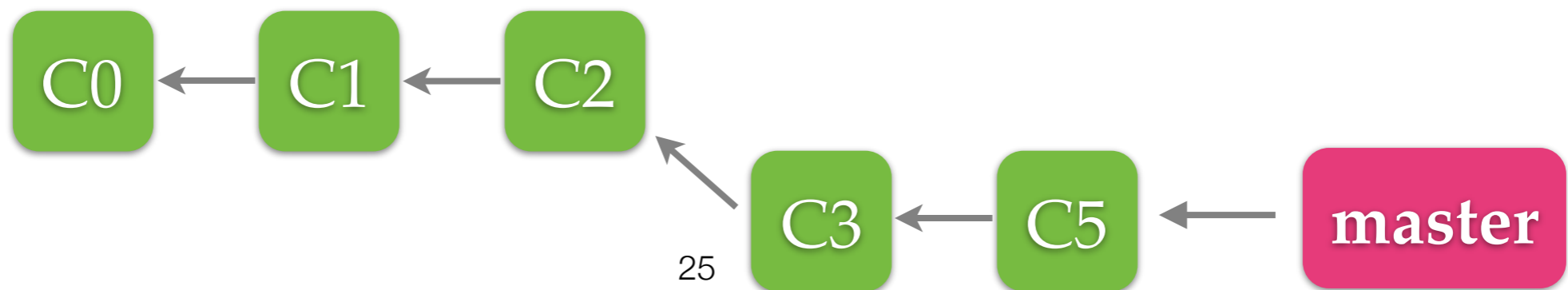


Merge

Working Area

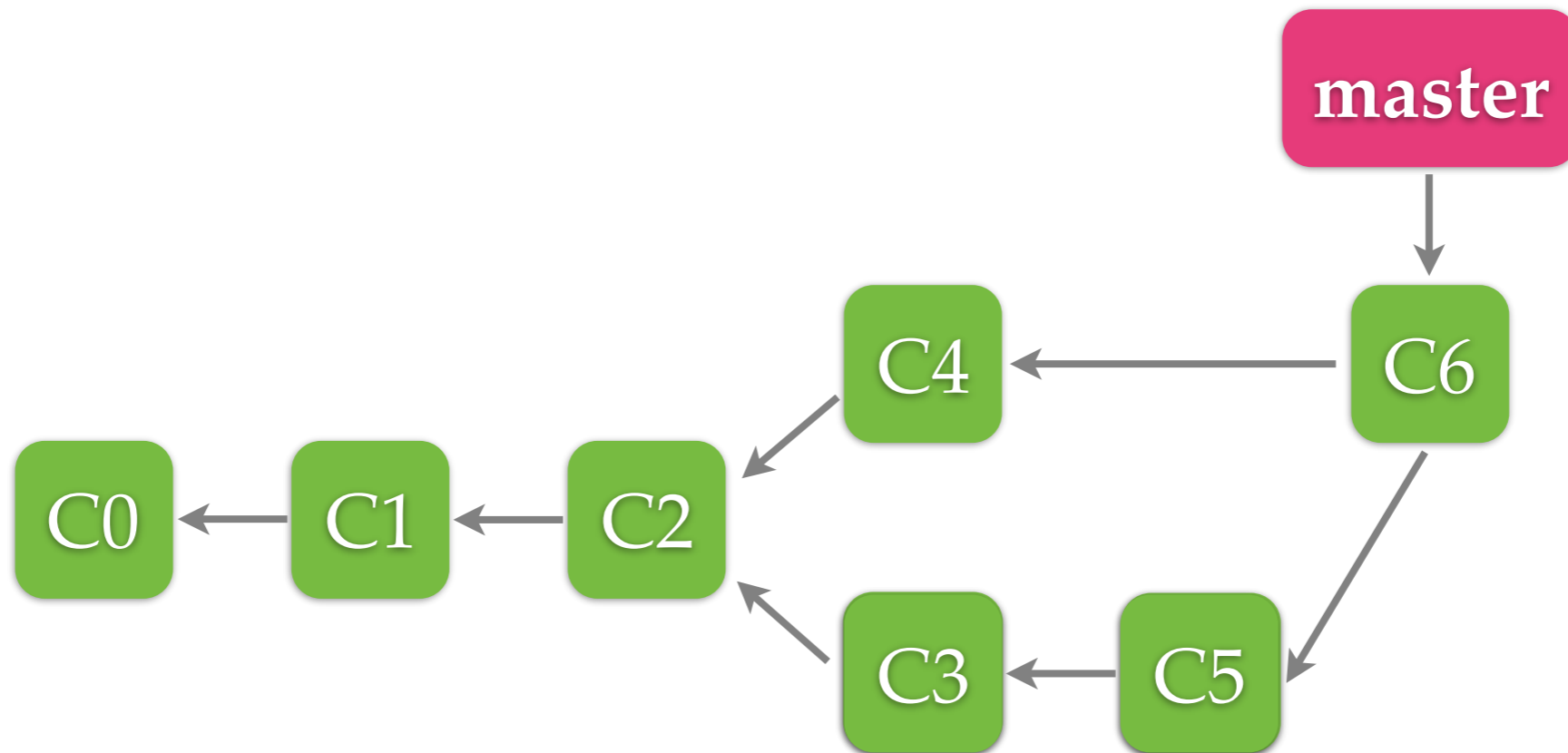


Repository

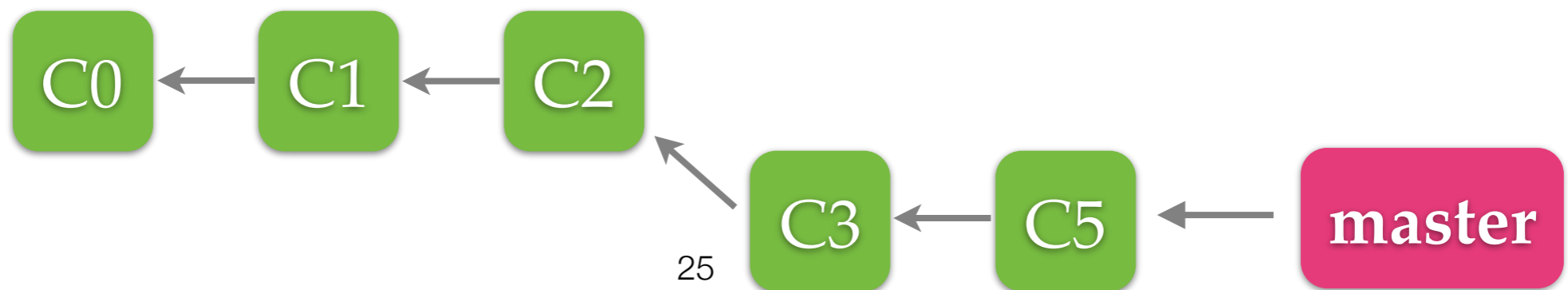


Merge

Working Area

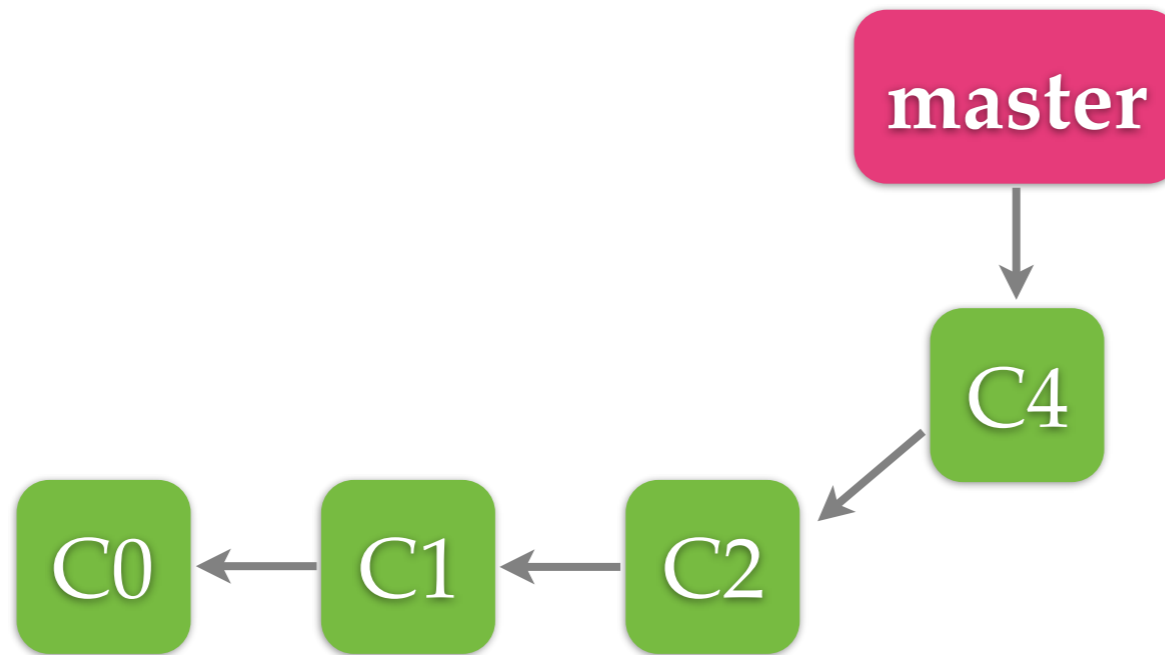


Repository

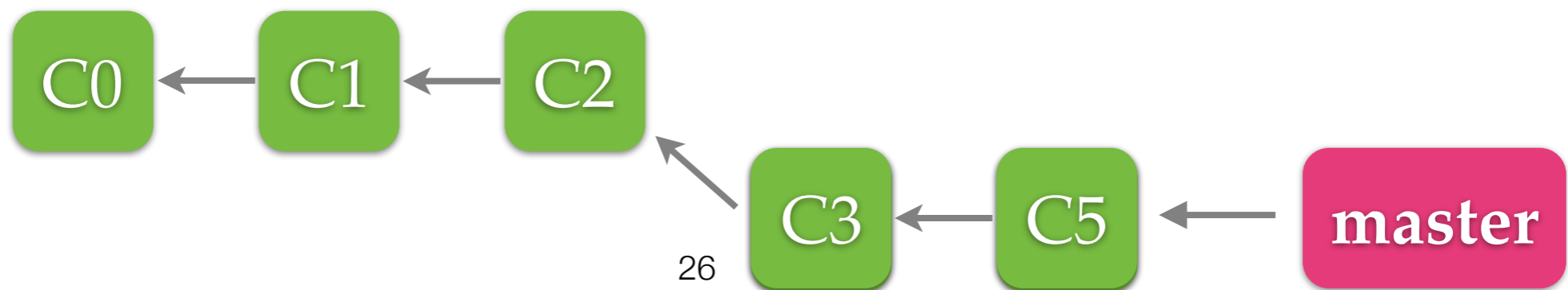


Rebase

Working Area

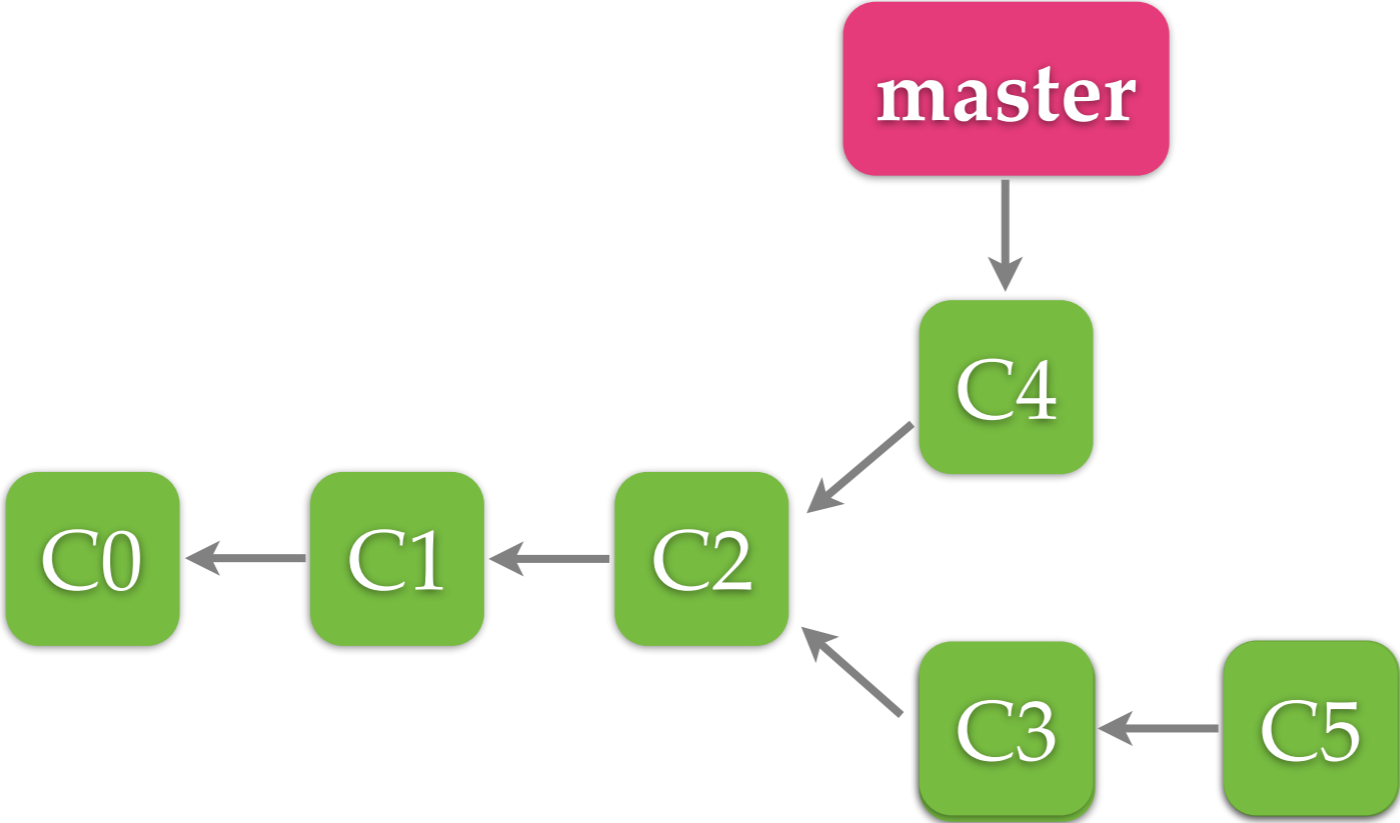


Repository

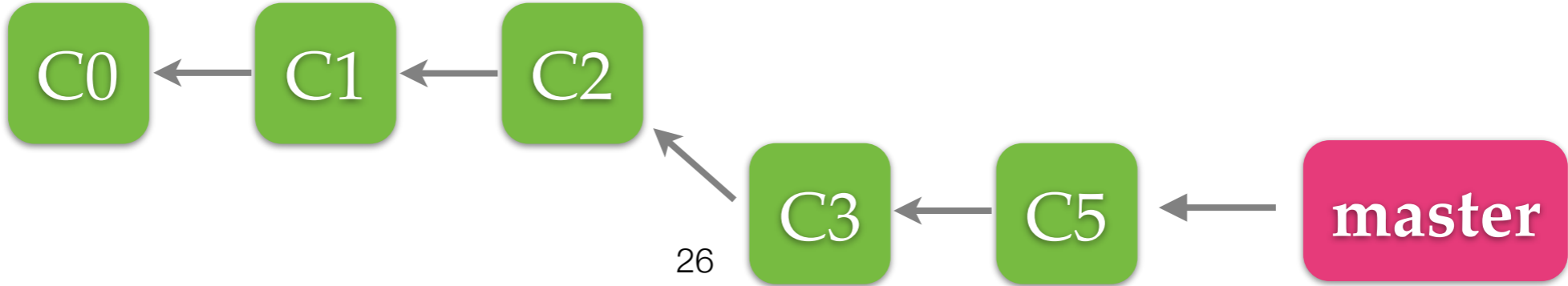


Rebase

Working Area

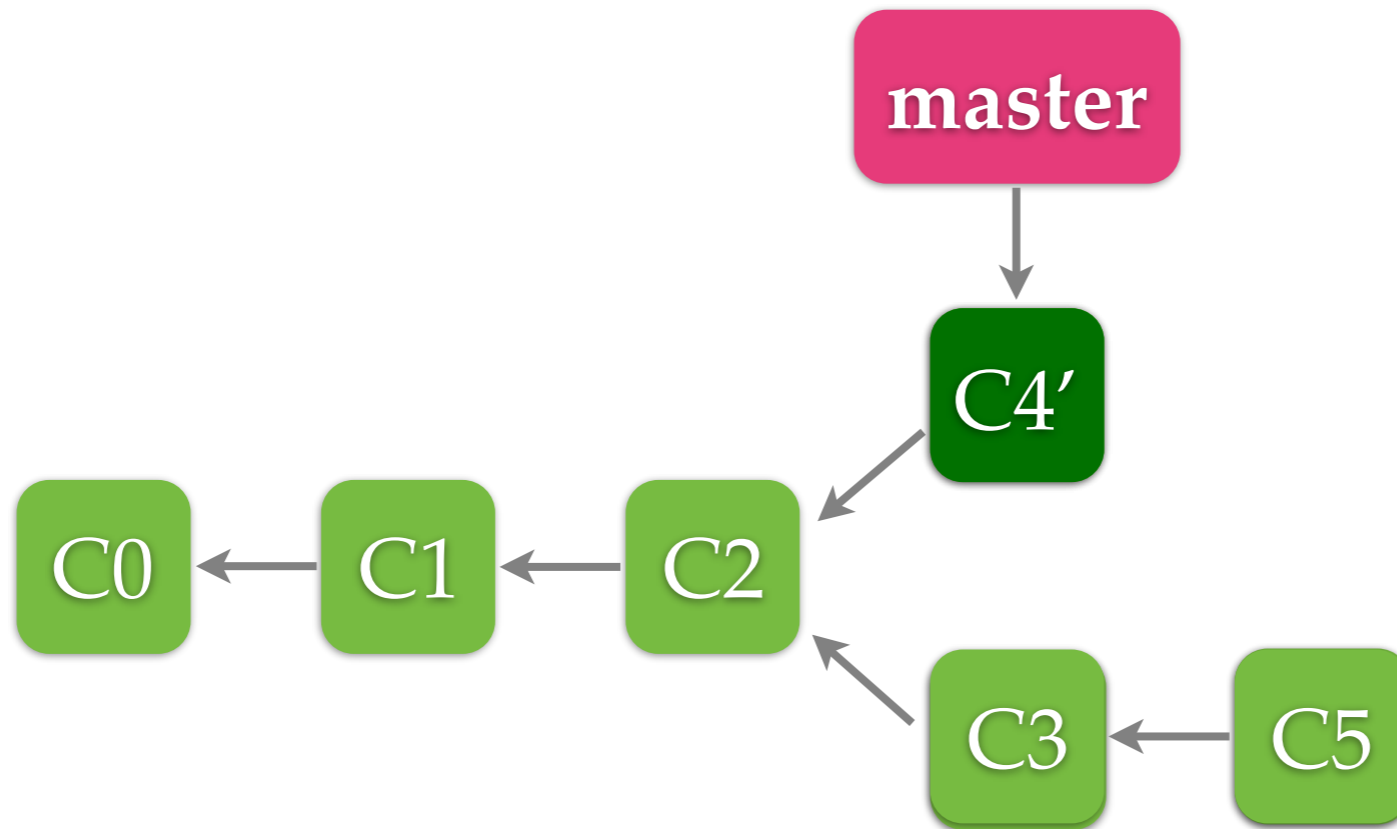


Repository

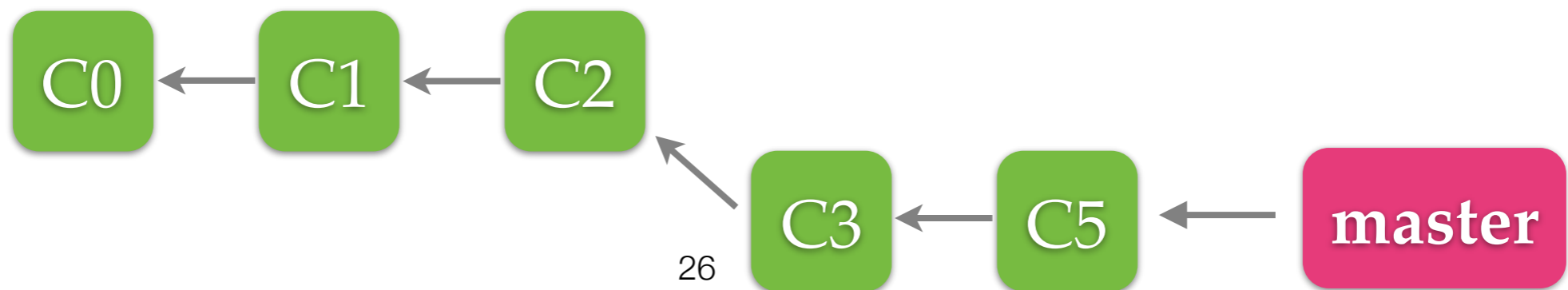


Rebase

Working Area



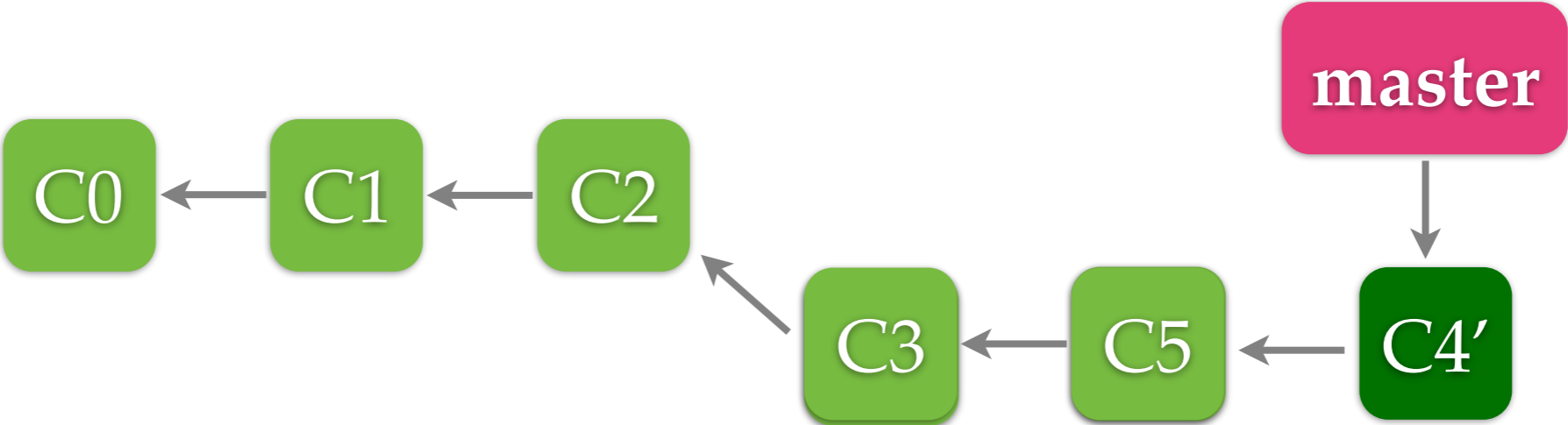
Repository



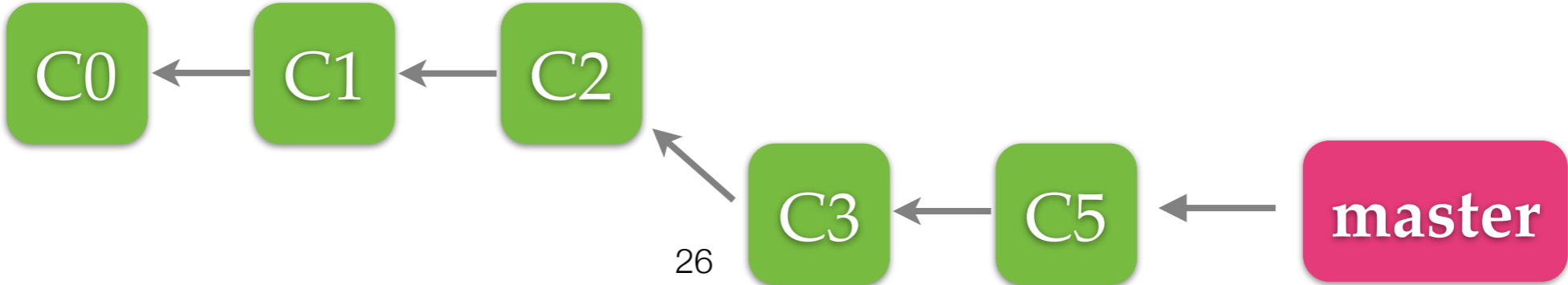
Rebase

Working Area

Fast Forward

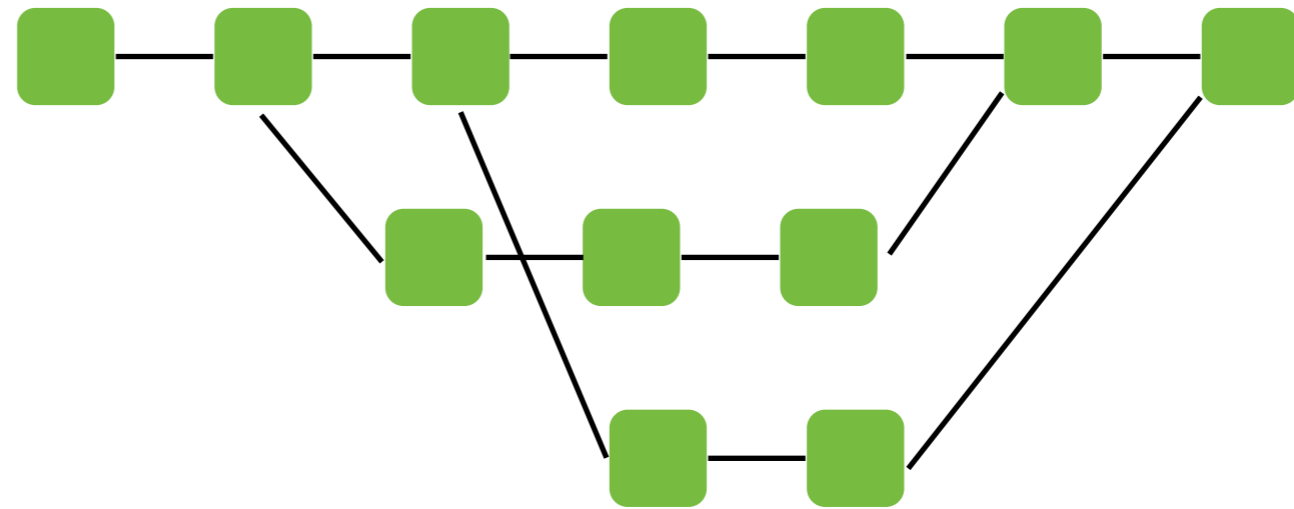


Repository



Merge VS Rebase

Merge



Rebase



Resolve Conflicts

```
DHCP-22126 project1 $ git fetch
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From /Users/mht208/demo/server
   3691b0d..57d4f6e  master    -> origin/master
DHCP-22126 project1 $ git merge
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
DHCP-22126 project1 $ █
```

Resolve Conflicts

1. Edit files with conflicts

```
█<<<<<<< HEAD
==== About Project A ====
=====
==== About Me ====
>>>>>>> refs/remotes/origin/master

This is readme.
```

Resolve Conflicts

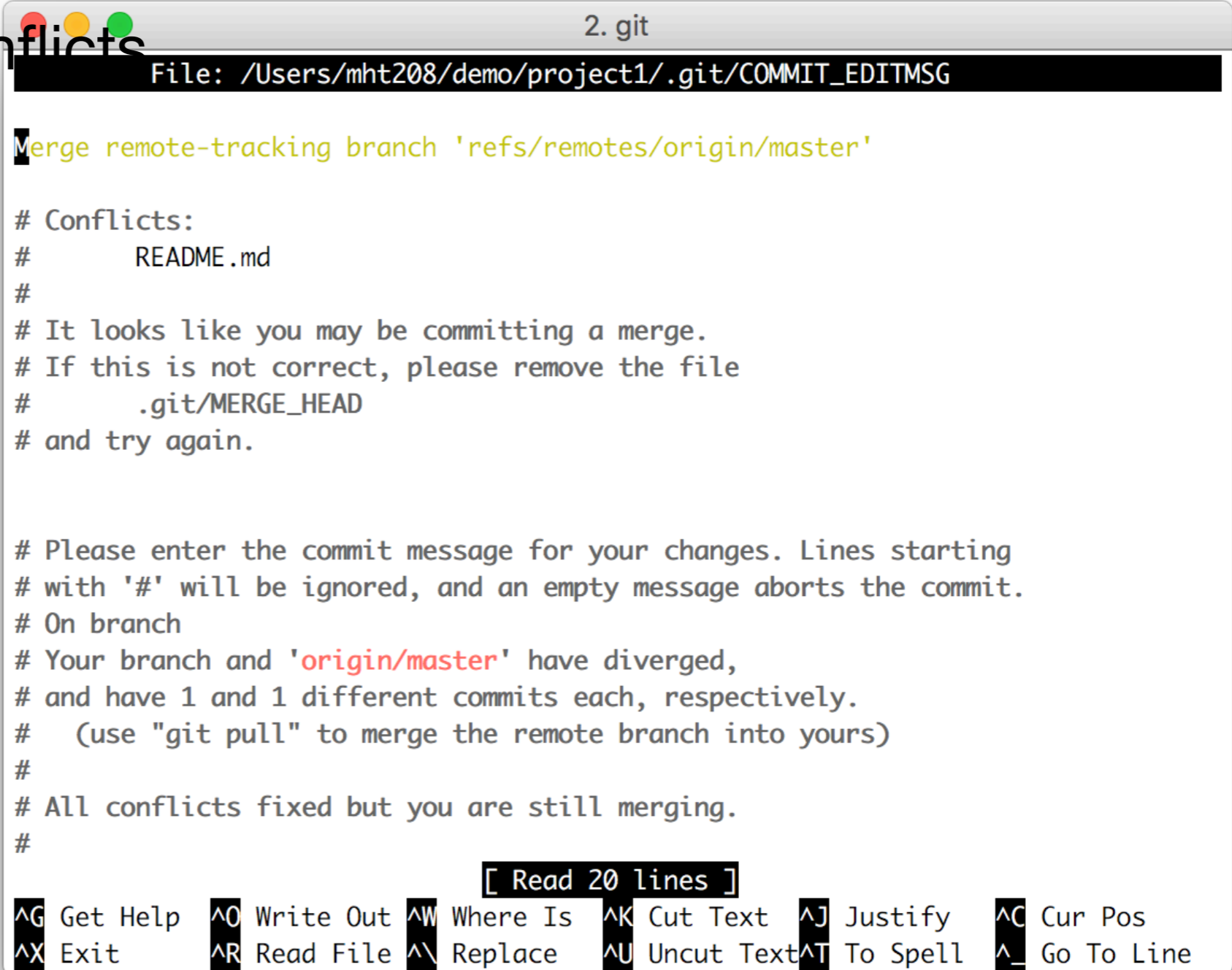
1. Edit files with conflicts
2. git add

```
█<<<<<<< HEAD
===== About Project A =====
=====
===== About Me =====
>>>>>>> refs/remotes/origin/master
```

This is readme.

Resolve Conflicts

1. Edit files with conflicts
2. git add
3. git commit



```
2. git
File: /Users/mht208/demo/project1/.git/COMMIT_EDITMSG
Merge remote-tracking branch 'refs/remotes/origin/master'

# Conflicts:
#   README.md
#
# It looks like you may be committing a merge.
# If this is not correct, please remove the file
#   .git/MERGE_HEAD
# and try again.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch
# Your branch and 'origin/master' have diverged,
# and have 1 and 1 different commits each, respectively.
# (use "git pull" to merge the remote branch into yours)
#
# All conflicts fixed but you are still merging.
#
[ Read 20 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```


Resolve Conflicts

1. Edit files with conflicts
2. git add
3. git commit

```
2. git
File: /Users/mht208/demo/project1/.git/COMMIT_EDITMSG
Merge remote-tracking branch 'refs/remotes/origin/master'

# Conflicts:
#   README.md
#
```

```
DHCP-22126 project1 $ git add README.md
```

```
DHCP-22126 project1 $ git commit
```

```
[master 3e5947f] Merge remote-tracking branch 'refs/remotes/origin/master'
```

```
DHCP-22126 project1 $
```


```
# with '#' will be ignored, and an empty message aborts the commit.
# On branch
# Your branch and 'origin/master' have diverged,
# and have 1 and 1 different commits each, respectively.
# (use "git pull" to merge the remote branch into yours)
#
# All conflicts fixed but you are still merging.
#
```

[Read 20 lines]

```
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

Resolve Conflicts

1. Edit files with conflicts
2. git add
3. git commit



```
2. git
File: /Users/mht208/demo/project1/.git/COMMIT_EDITMSG
Merge remote-tracking branch 'refs/remotes/origin/master'

# Conflicts:
#   README.md
#
```

```
DHCP-22126 project1 $ git add README.md
```

```
DHCP-22126 project1 $ git commit
```

```
[master 3e5947f] Merge remote-tracking branch 'refs/remotes/origin/master'
```

```
DHCP-22126 project1 $
```

```
# with '#' will be ignored, and an empty message aborts the commit.
# On branch
```

```
* 3e5947f (HEAD -> master) Merge remote-tracking branch 'refs/remotes/origin/master'
```

```
| \
```

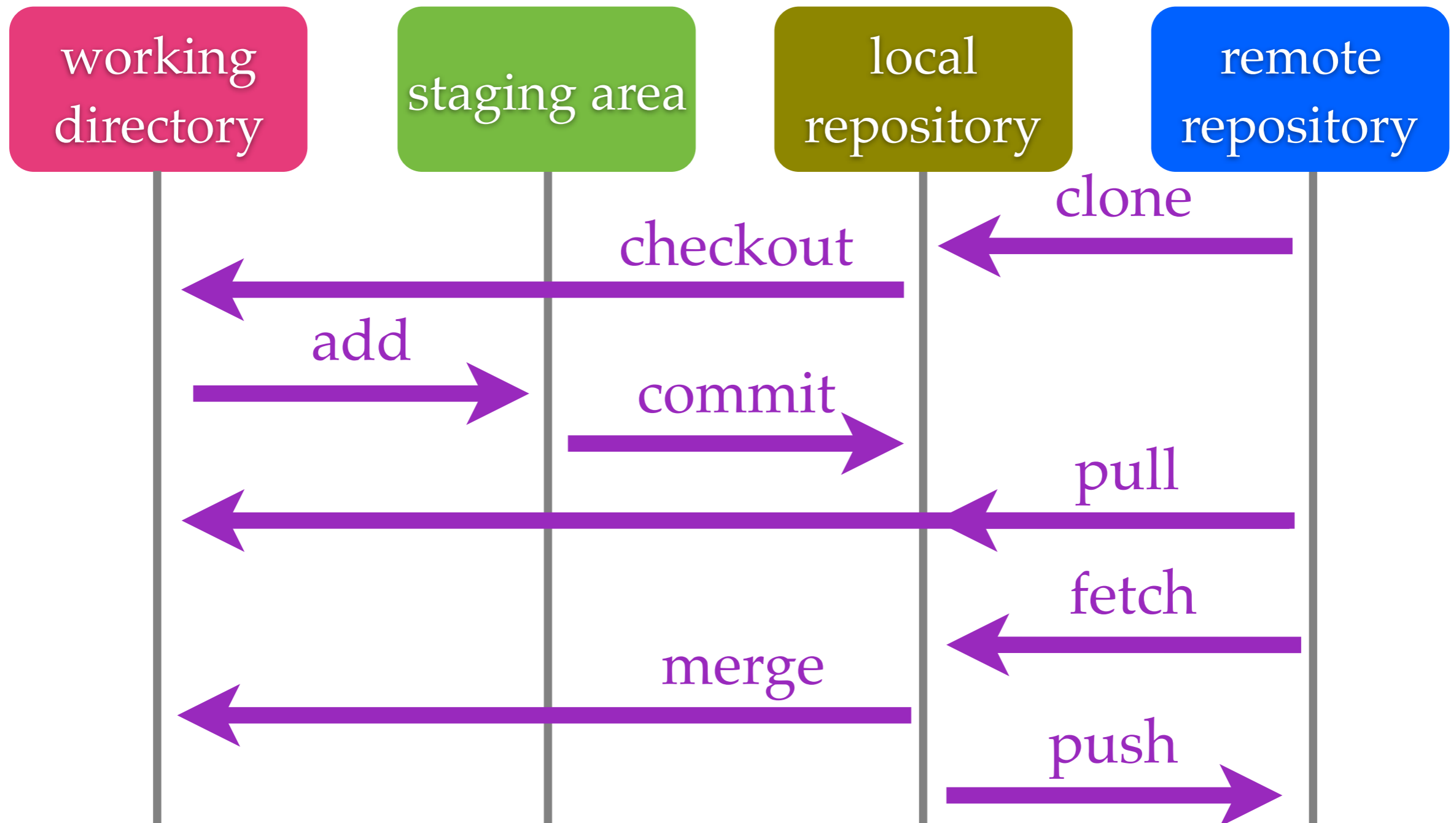
```
| * 57d4f6e (origin/master) Modify README.md.
```

```
* | 22804d8 Modify README.md.
```

```
| /
```

```
* 3691b0d Add README.md.
```

Summary



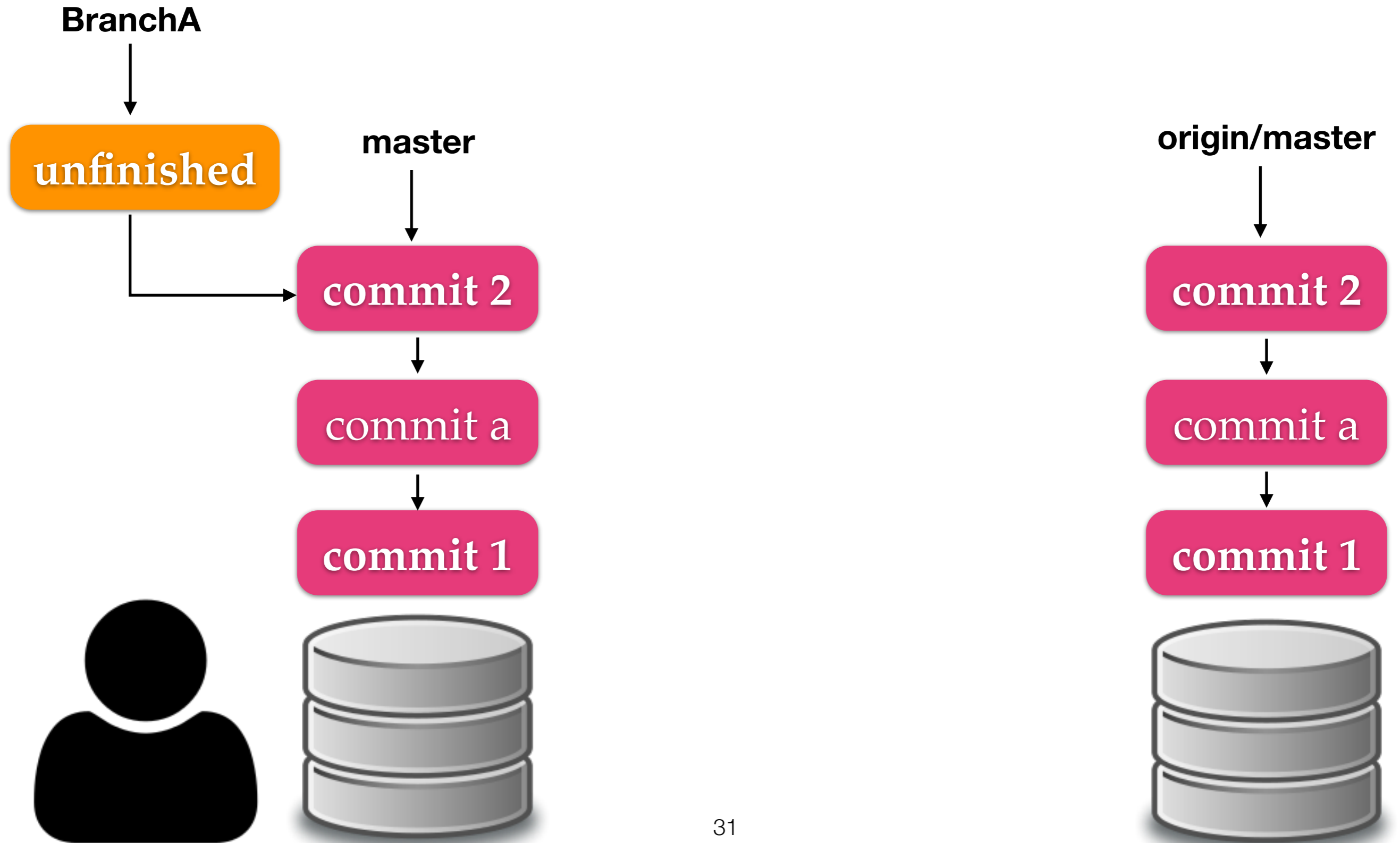
Distributed Version Control

- BitKeeper
- Git (2005)
- Mercurial (2005)

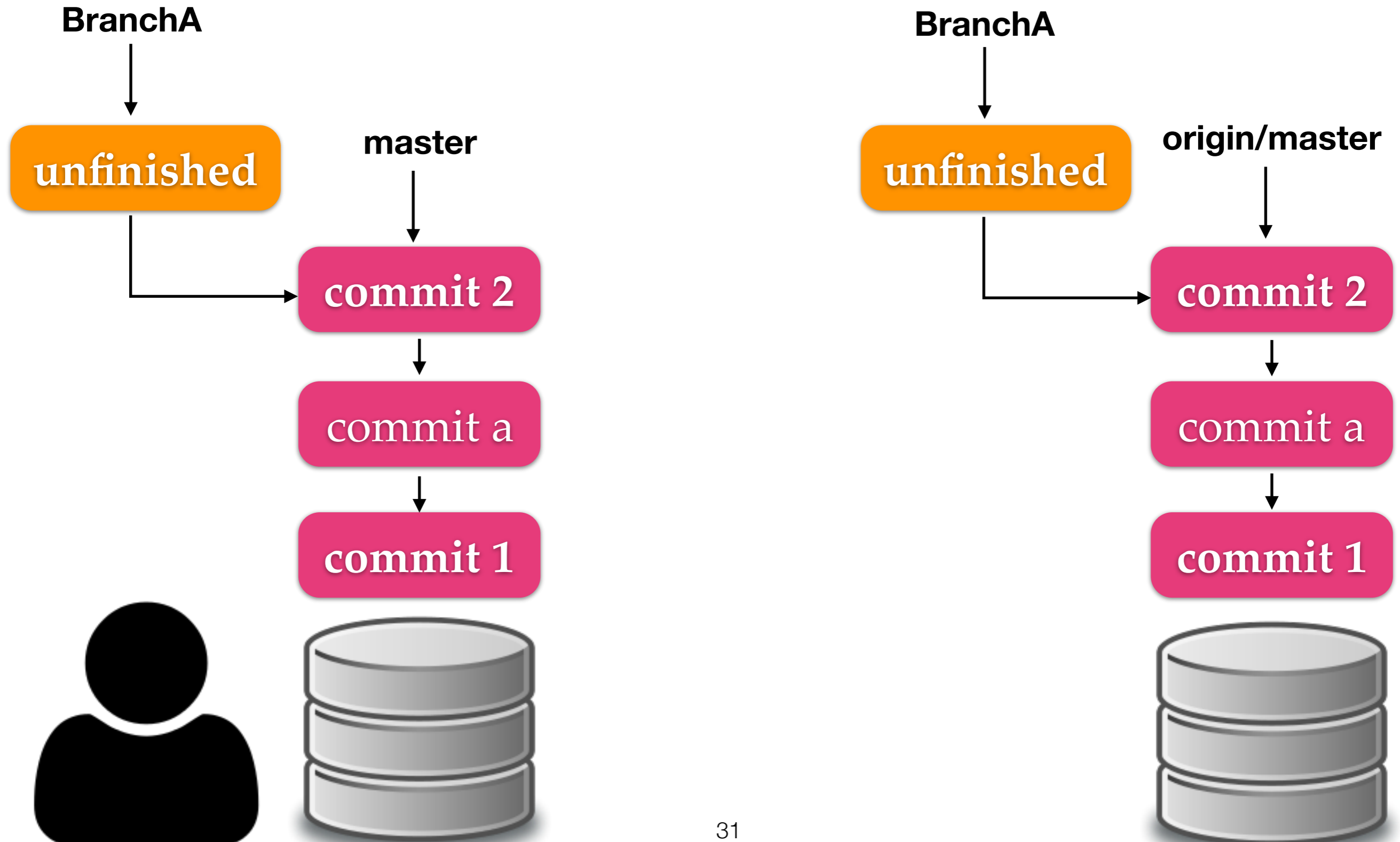
Branching



Branching



Branching



Branching

BranchA

BranchA

unfinished

master

unfinished

origin/master

```
git branch BranchA: create a branch named BranchA
git push origin BranchA: push BranchA to server
git checkout BranchA: switch to BranchA
```

commit 2

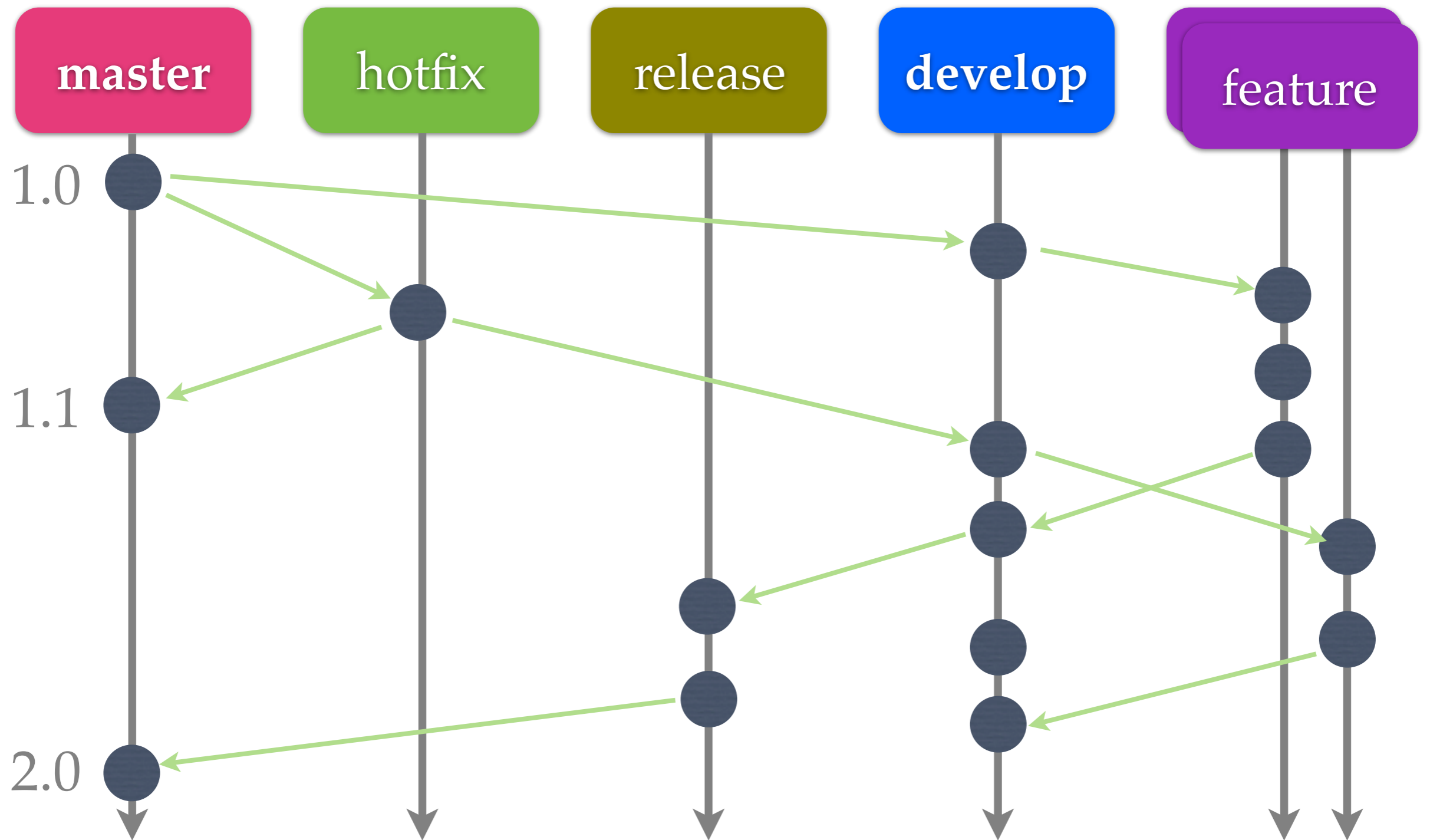
commit a

commit 1

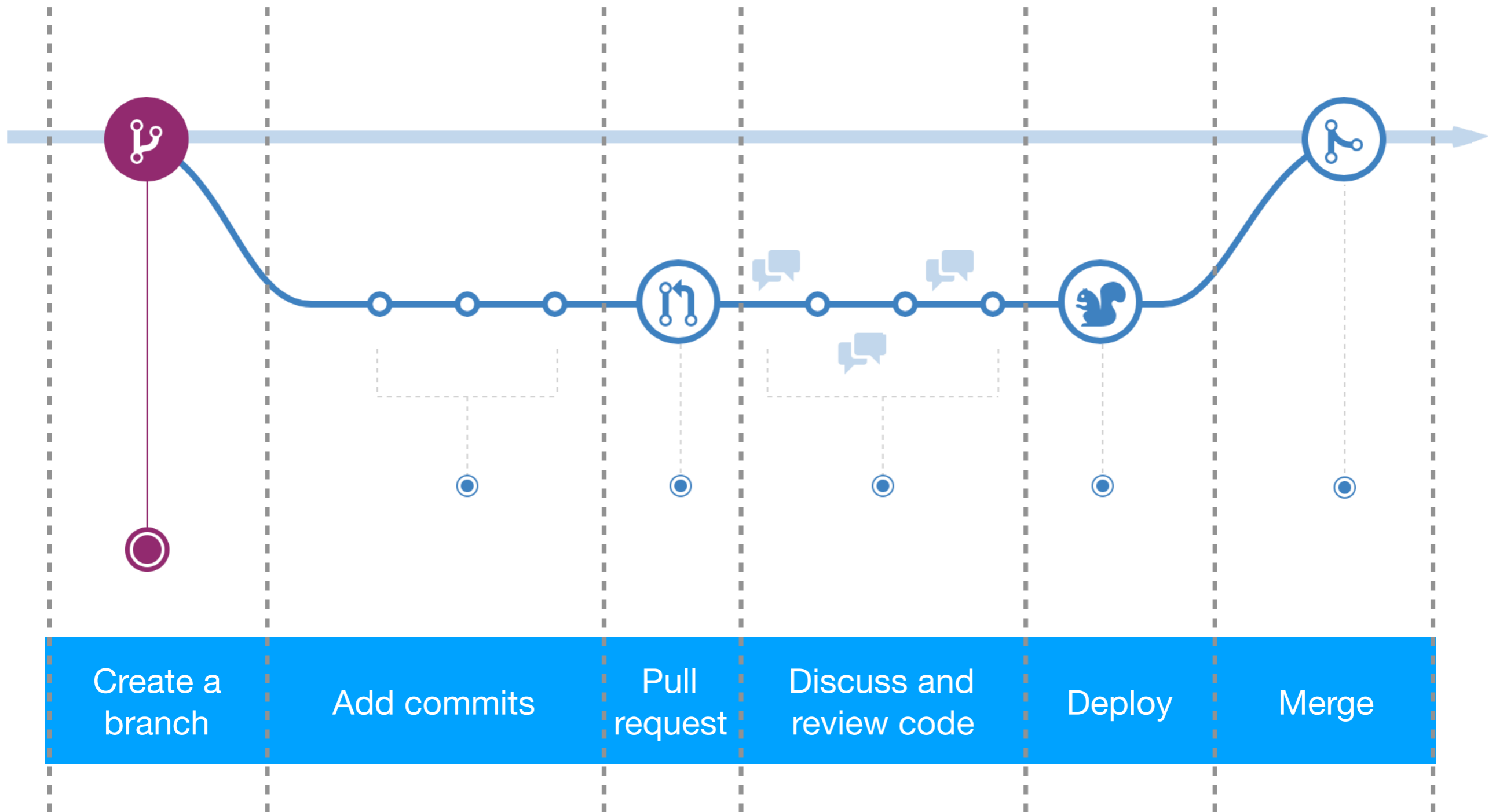
commit 1



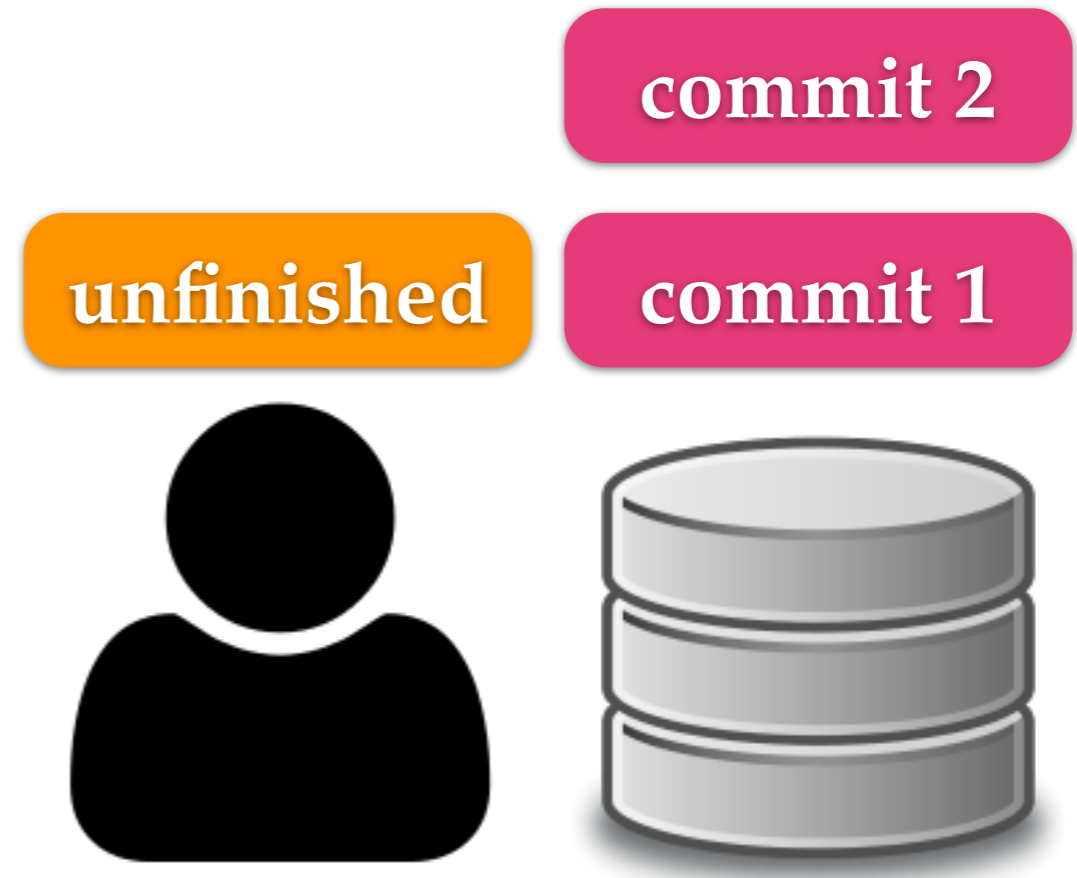
Git Flow



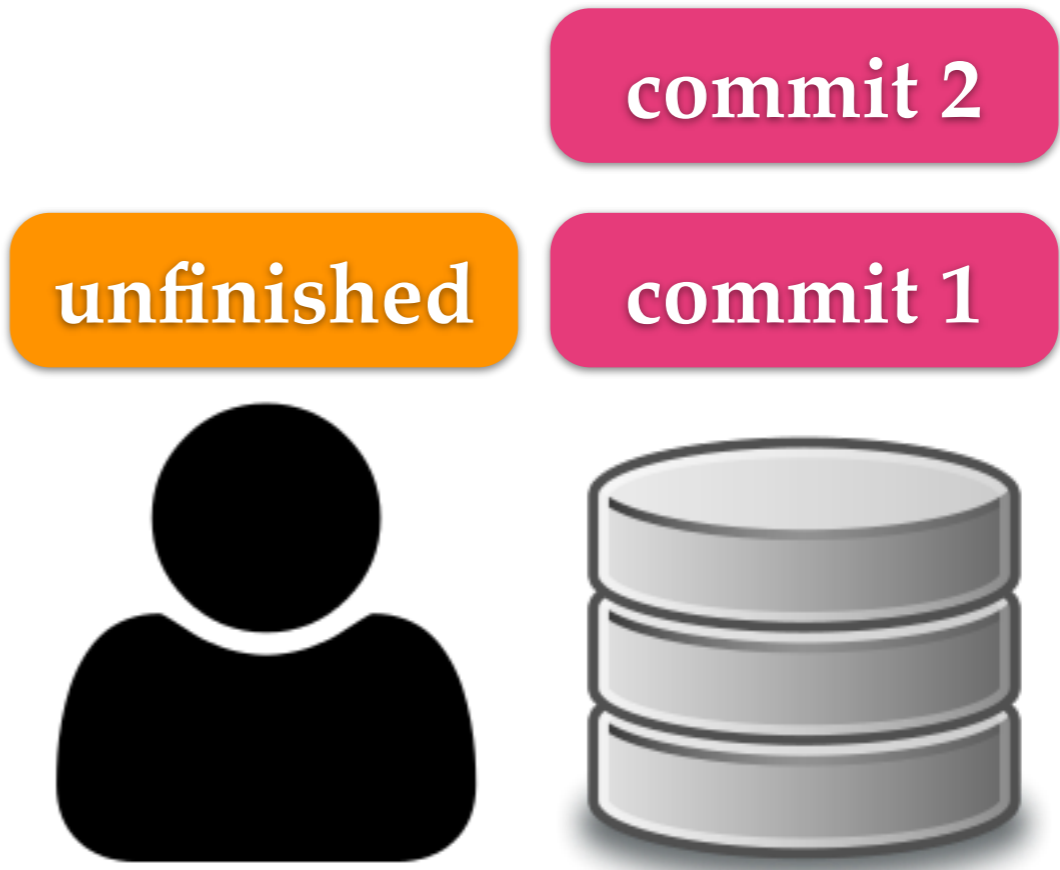
GitHub Flow



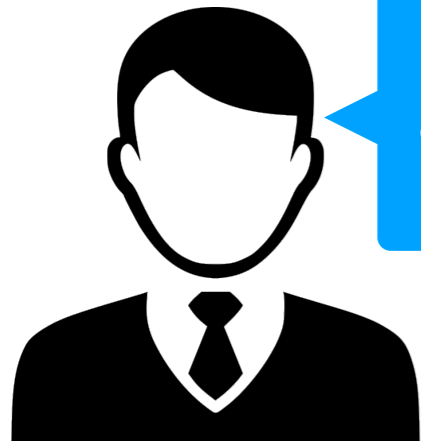
Stash



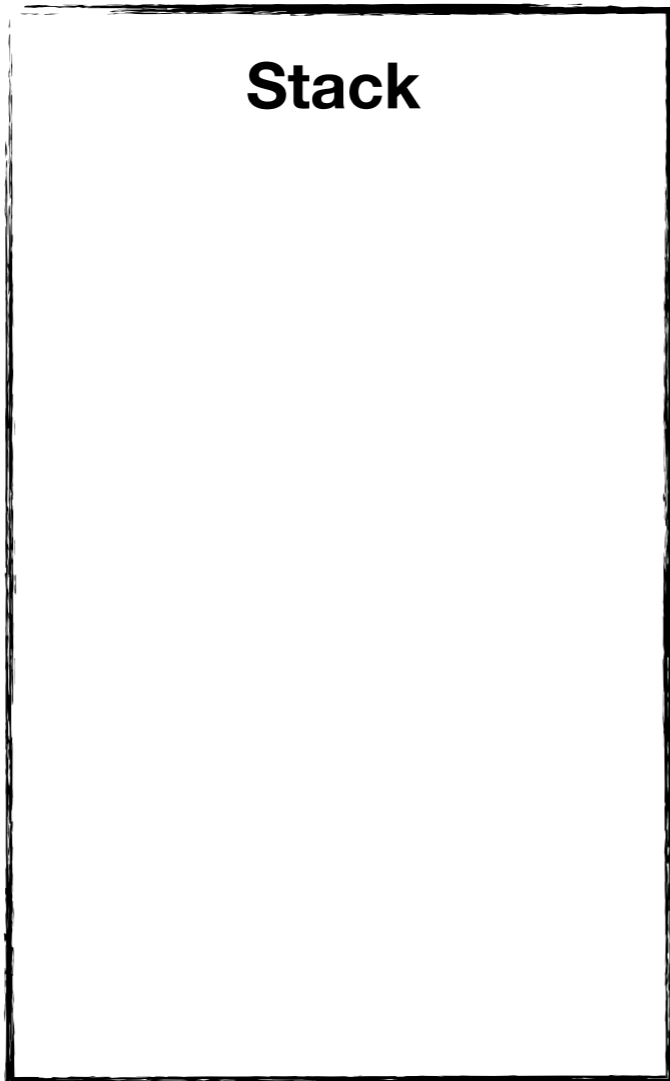
Stash



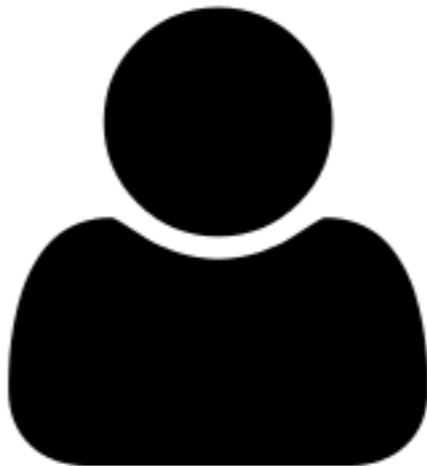
Stash



add feature A



unfinished

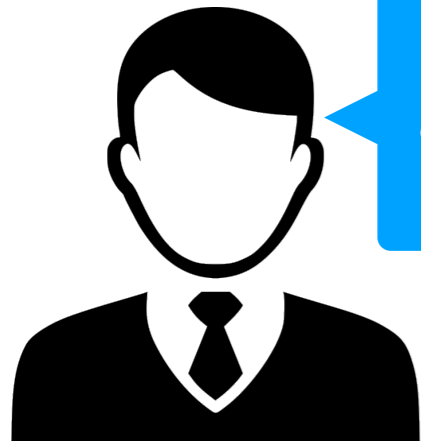


commit 2

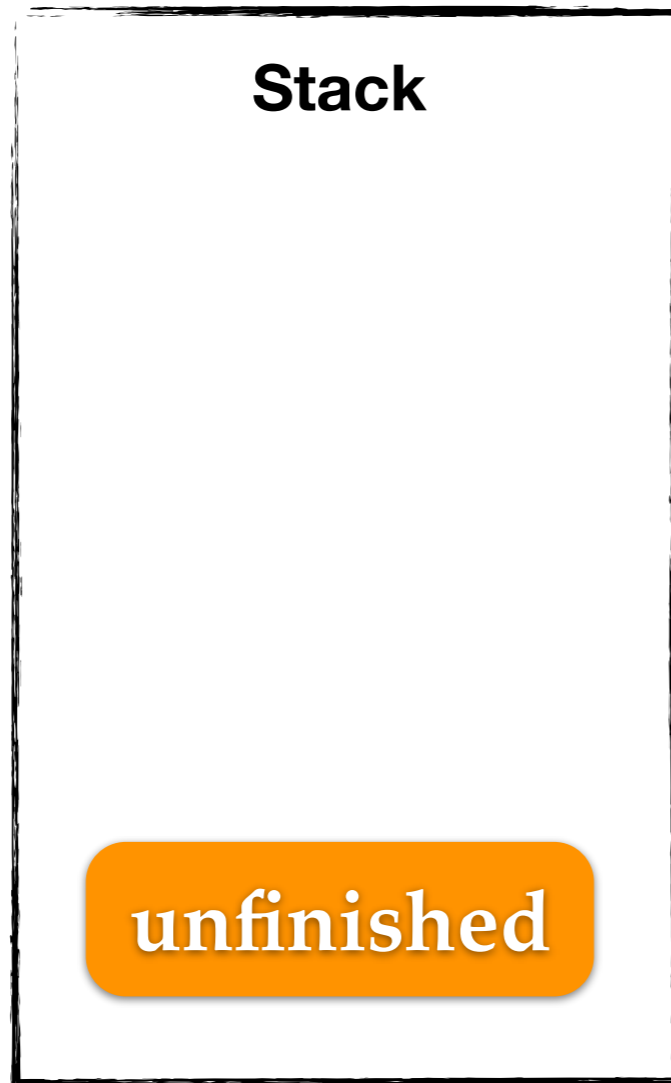
commit 1



Stash



add feature A



feature A

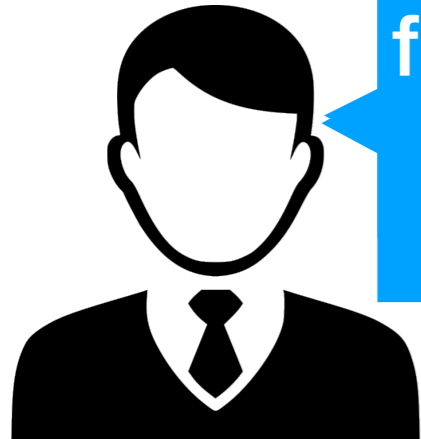


commit 2

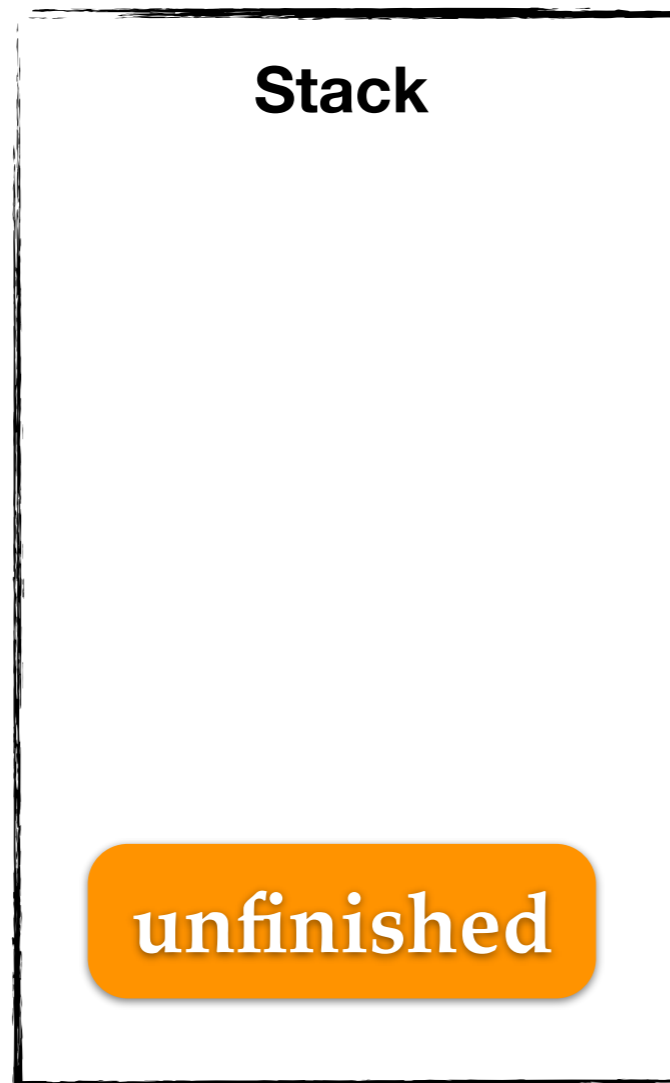
commit 1



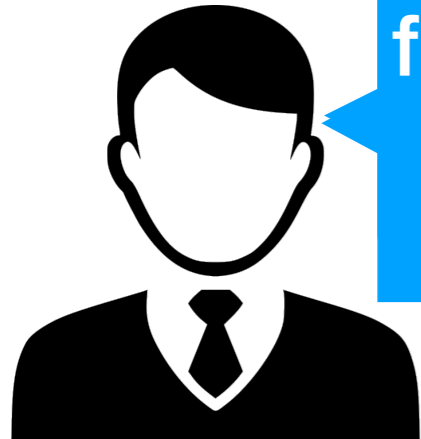
Stash



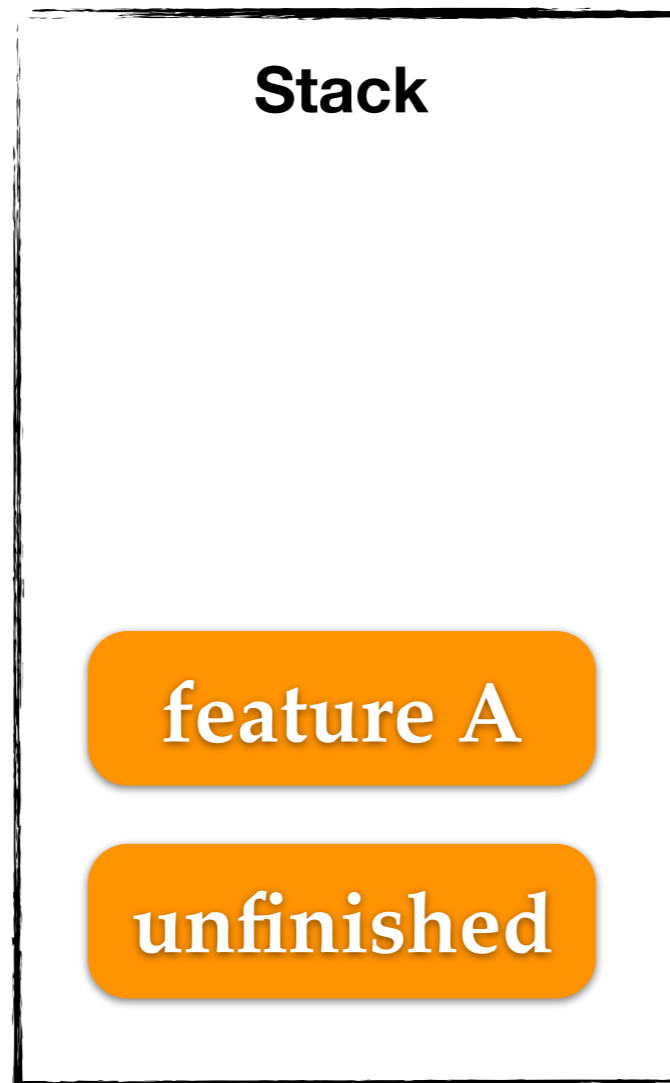
fix a bug, right now!



Stash



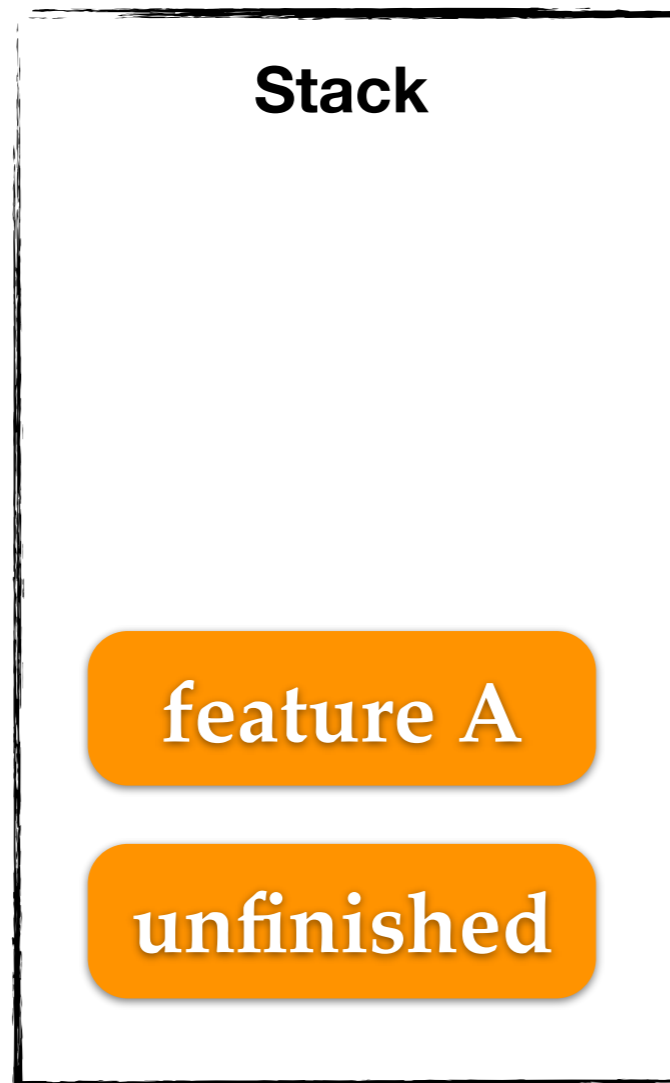
fix a bug, right now!



Stash



fix a bug, right now!



bug fix

commit 2

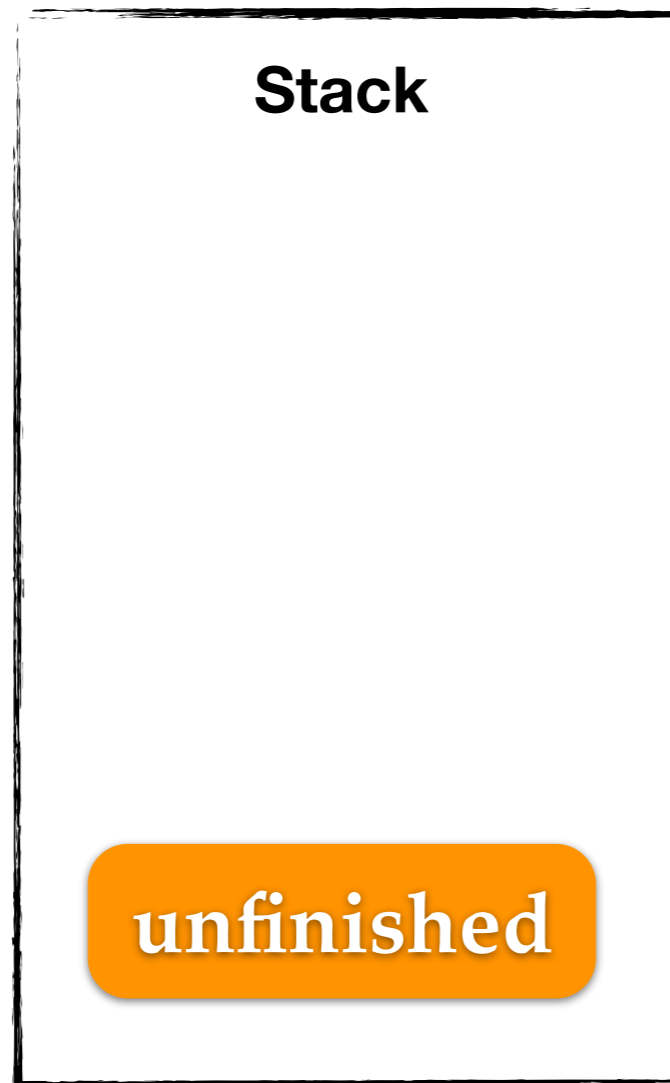
commit 1



Stash



fix a bug, right now!



feature A'



bug fix

commit 2

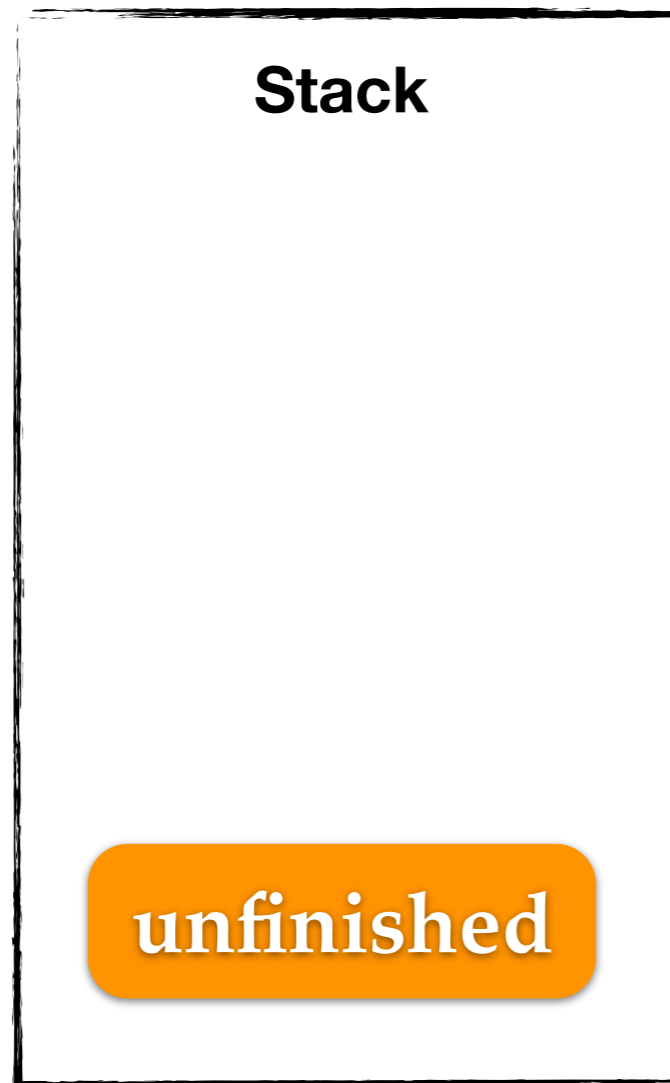
commit 1



Stash



fix a bug, right now!



feature A'

bug fix

commit 2

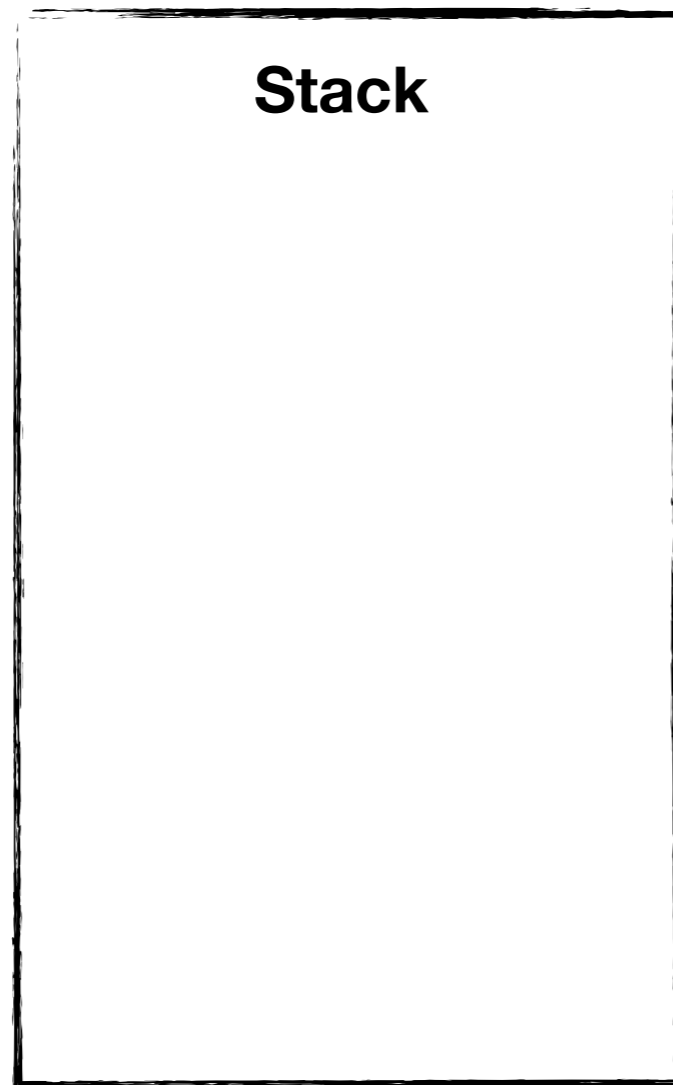
commit 1



Stash



fix a bug, right now!



unfinished'



feature A'

bug fix

commit 2

commit 1



Squash

```
DHCP-22126 squash $ git glog
* 5718716 (HEAD -> master) Improve performance.
* e5abbde Fix a bug in function B.
* be689c1 Add test cases for functions A and B.
* 4960ef6 Fix a bug in function A.
* 697a699 Add function B.
* d7bce04 Add function A.
* aaa944b Add main.c.
DHCP-22126 squash $ █
```



glog is an alias of “log --graph --decorate --pretty=oneline --abbrev-commit”

Squash

```
DHCP-22126 squash $ git log
* 5718716 (HEAD -> master) Improve performance.
* e5abbde Fix a bug in function B.
* be689c1 Add test cases for functions A and B.
* 4960ef6 Fix a bug in function A.
* 697a699 Add function B.
* d7bce04 Add function A.
* aaa944b Add main.c.
DHCP-22126 squash $ █
```



```
DHCP-22126 squash $ git rebase -i HEAD~6
HEAD: the version of the working tree
```

Squash

```
File: /Users/mht208/demo/squash/.git/rebase-merge/git-rebase-todo
```

```
pick d7bce04 Add function A.  
pick 697a699 Add function B.  
pick 4960ef6 Fix a bug in function A.  
pick be689c1 Add test cases for functions A and B.  
pick e5abbde Fix a bug in function B.  
pick 5718716 Improve performance.
```

```
# Rebase aaa944b..5718716 onto aaa944b (6 commands)  
#  
# Commands:  
# p, pick = use commit  
# r, reword = use commit, but edit the commit message  
# e, edit = use commit, but stop for amending  
# s, squash = use commit, but meld into previous commit  
# f, fixup = like "squash", but discard this commit's log message  
# x, exec = run command (the rest of the line) using shell  
# d, drop = remove commit  
#  
# These lines can be re-ordered; they are executed from top to bottom.  
#
```

Squash

File: /Users/mht208/demo/squash/.git/rebase-merge/git-rebase-todo Modified

```
pick d7bce04 Add function A.
squash 697a699 Add function B.
squash 4960ef6 Fix a bug in function A.
squash be689c1 Add test cases for functions A and B.
squash e5abbde Fix a bug in function B.
squash 5718716 Improve performance.
```

```
# Rebase aaa944b..5718716 onto aaa944b (6 commands)
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
# d, drop = remove commit
#
# These lines can be re-ordered; they are executed from top to bottom.
#
```


Squash

```
GNU nano 2.8.5 File: /Users/mht208/demo/squash/.git/COMMIT_EDITMSG
```

```
# This is a combination of 6 commits.
```

```
# This is the 1st commit message:
```

```
Add function A.
```

```
# This is the commit message #2:
```

```
Add function B.
```

```
# This is the commit message #3:
```

```
Fix a bug in function A.
```

```
# This is the commit message #4:
```

```
Add test cases for functions A and B.
```

```
# This is the commit message #5:
```

```
Fix a bug in function B.
```

Squash

```
GNU nano 2.8.5 File: /Users/mht208/demo/squash/.git/COMMIT_EDITMSG Modified
```

Add feature A. █

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Sep 13 09:36:38 2017 +0800
#
# interactive rebase in progress; onto aaa944b
# Last commands done (6 commands done):
#   squash e5abbde Fix a bug in function B.
#   squash 5718716 Improve performance.
# No commands remaining.
```

Squash

```
DHCP-22126 squash $ git rebase -i HEAD~6
[detached HEAD 4baf53e] Add feature A.
Date: Wed Sep 13 09:36:38 2017 +0800
1 file changed, 12 insertions(+)
Successfully rebased and updated refs/heads/master.
DHCP-22126 squash $ git glog
* 4baf53e (HEAD -> master) Add feature A.
* aaa944b Add main.c.
DHCP-22126 squash $ █
```



.gitignore

```
DHCP-22126 project1 $ git status
```

```
On branch master
```

```
Your branch is ahead of 'origin/master' by 2 commits.
```

```
(use "git push" to publish your local commits)
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
A.c
```

```
A.o
```

```
B.c
```

```
B.o
```

```
C.class
```

```
C.java
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

.gitignore

```
DHCP-22126 project1 $ cat .gitignore
```

```
*.o
```

```
*.class
```

```
DHCP-22126 project1 $ git status
```

```
On branch master
```

```
Your branch is ahead of 'origin/master' by 2 commits.
```

```
(use "git push" to publish your local commits)
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
  .gitignore
```

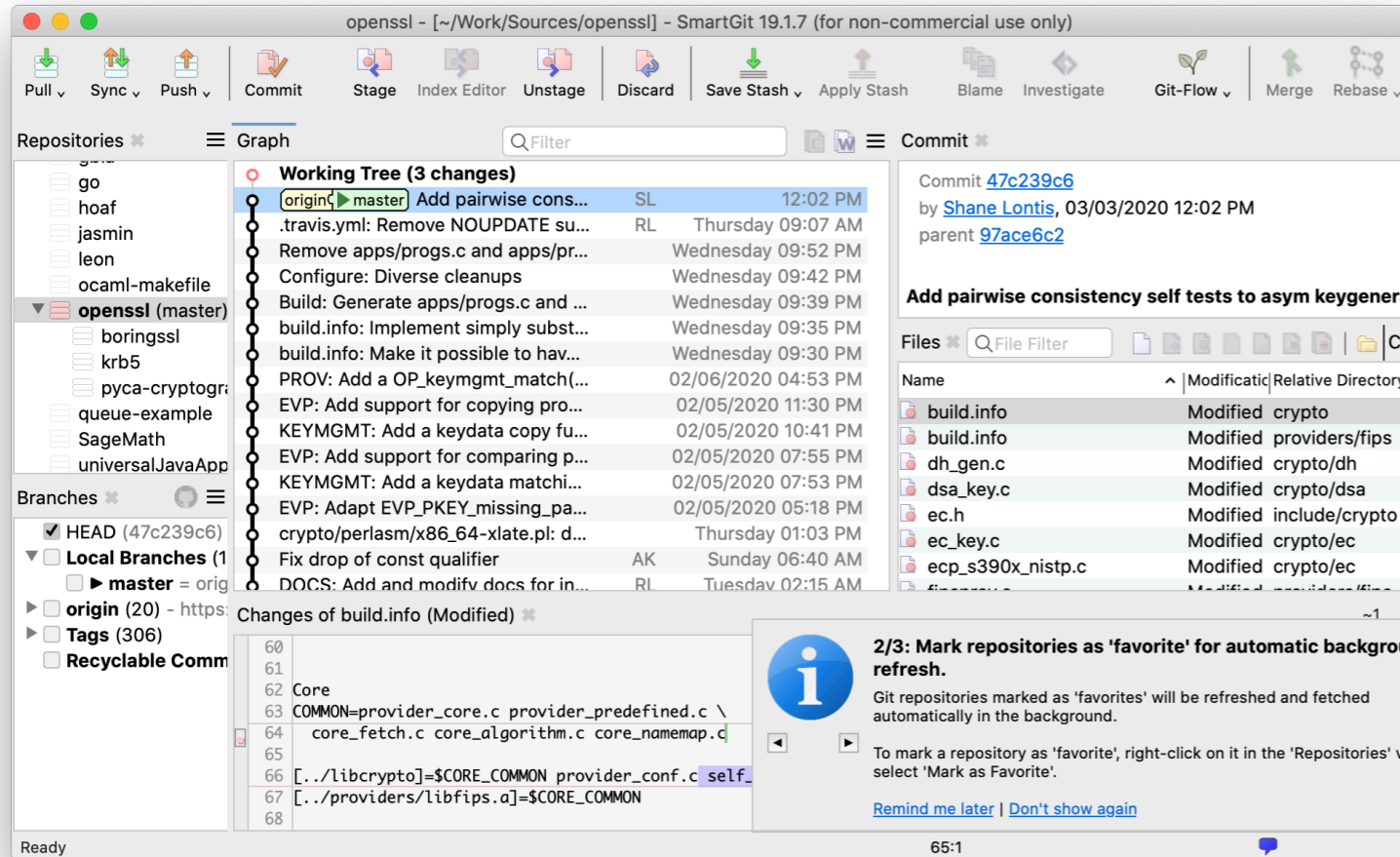
```
  A.c
```

```
  B.c
```

```
  C.java
```

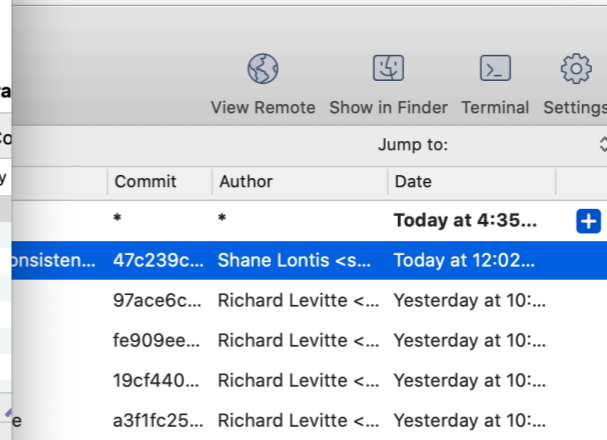
```
nothing added to commit but untracked files present (use "git add" to track)
```

GUI Clients



SmartGit

SourceTree



GitHub Desktop

References

- <http://git-scm.com/book>
- <http://git-scm.com/docs>